

RESEARCH

Open Access



# A virtual grid-based real-time data collection algorithm for industrial wireless sensor networks

Chuan Zhu, Xiaohan Long, Guangjie Han<sup>\*</sup> , Jinfang Jiang and Sai Zhang

## Abstract

Industrial wireless sensor networks (IWSNs) have been widely used in many application scenarios, and data collection is an extremely significant part of IWSNs. Moreover, a mobile sink is widely used in industrial wireless sensor networks to collect sensory data and alleviate the “hot spot” problem effectively. However, usage of a mobile sink introduces some challenges, such as updating of a mobile sink’s location and planning of a mobile sink’s trajectory. Meanwhile, the impact of different distribution types of events on data collection has not been sufficiently valued in designing of data collection algorithm for IWSNs yet. To overcome these challenges, a virtual grid-based real-time data collection algorithm for applications with centrally distributed events (VGDC-A-C) is proposed in this paper to gain a reliable data gathering for IWSNs. In the target application scenarios, the events are distributed centrally, so we mainly focus on how to shorten the routing paths and decrease the transmission delay. In our VGDC-A-C, a mobile sink can adjust its movement dynamically according to the changes in event areas. The adjustment of a mobile sink movement strategy includes two aspects. The first one is the dynamic adjustment of a mobile sink’s parking time, and the second one denotes the moving toward event area of a mobile sink. Thus, a mobile sink adjusts its location such that it can get closer to the event area. Hence, the total length of routing is getting shorter so that source nodes can upload sensory data faster. Analysis and simulation results show that compared with the existing work, our VGDC-A-C increases the network lifetime and decreases transmission delay.

**Keywords:** Industrial wireless sensor networks, Data collection, Mobile sink, Centrally distributed events

## 1 Introduction

With the development of industrial wireless communication technologies, microelectronics, sensors, distributed information processing, and embedded computers, the industrial wireless sensor networks (IWSNs) have been widely used in many application scenarios such as poisonous gas boundary detection [1, 2], pollution monitoring [3–5], and production monitoring [6, 7]. Data collected by sensor nodes need to be uploaded to a sink quickly and accurately via data routing to the sink. The stability and accuracy of data collection are the guarantees of IWSNs’ normal operations. Therefore, the data collection and routing [8, 9] plays an important role in IWSNs. In the traditional IWSNs, a static sink or a base station is

used to collect sensory data from sensor nodes whose batteries can be charged in some scenarios [10], and all source nodes deliver sensory data to the static sink by via multi-hop transmission [11]. This way of data collection always leads to a “hot spot” problem [12] that means nodes near the sink or base station run out of energy very fast so that the network performance has been affected. Due to that, a mobile sink is introduced to solve this problem. A mobile sink can alleviate “hot spot” problem efficiently. Namely, plenty of researchers have revealed that mobile sink can make the data collection energy efficient [13–27]. In their researches, intelligent unmanned vehicle or unmanned aerial vehicle is appointed to be a mobile sink, and all these vehicles are equipped with the industrial wireless communication device and data processor. A mobile sink can move to the locations of source nodes and collect sensory data directly from the source nodes.

\*Correspondence: [hanguangjie@gmail.com](mailto:hanguangjie@gmail.com)

Department of Internet of Things Engineering, Hohai University, Changzhou, China

Moreover, a mobile sink can balance the energy consumption and prolong the lifetime of the network. However, the use of a mobile sink introduces two new challenges in the data collection of IWSNs. The first one is the way the mobile sink's latest location is updated. The traditional way of this updating is that a mobile sink broadcasts the updated information on its location to the entire network. However, a frequent broadcast may lead to high overheads and shorten the lifetime of a network. Hence, it is challenging to find a suitable way to update the information of mobile sink with lower overheads. Another challenge is the way the trajectory of a mobile sink is planned [28]. As the sensory data are delivered to a mobile sink through multi-hops, the sensor nodes around the mobile sink run out of energy very fast. Moreover, the number of transmission hops affects the transmission delay. Hence, the trajectory of a mobile sink affects the balance of energy consumption and transmission delay greatly.

Meanwhile, the impact of different distribution types of events on data collection has not been sufficiently valued in designing of data collection algorithms for IWSNs. In current studies, source nodes are always distributed evenly in the network. However, in some application scenarios such as industrial fire detection, the monitored targets are distributed in a local area. Therefore, here, we focus on the scenarios wherein the source nodes are distributed centrally. Taking into consideration the challenges mentioned above and the distribution types of source nodes, we propose a virtual grid-based real-time data collection algorithm for applications with centrally distributed events for industrial wireless sensor networks.

The contributions of this paper are summarized as follows. A real-time mobile data collection algorithm based on a virtual grid structure VGDCa-C is proposed. By constructing a virtual grid structure in the network, the information on a mobile sink location can be updated locally such that the energy consumption of a mobile sink and data transmission delay can be reduced and network lifetime can be extended. The algorithm proposed in this paper is suitable for scenarios wherein events are centrally distributed, such as the monitoring of a fire in the industrial factories or malfunction monitoring of industrial equipment.

The remainder of this paper is organized as follows. Firstly, the related works of data collection algorithms with a mobile sink are presented in Section 2. The details of the VGDCa-C are described in Section 3. In Section 4, the simulation experiments and performance evaluations are provided. A brief conclusion is given in Section 5, and abbreviations used in the manuscript are listed in the "Abbreviation" section.

## 2 Related works

### 2.1 Overview

The VGDCa-C is a data collection algorithm based on a virtual structure, which compensates the lack of data collection algorithms for non-virtual structures. However, the trajectory of a mobile sink and the corresponding location updating also need to be taken into consideration. Recently proposed related algorithms can be classified into two following categories: (1) non-virtual structure-based data collection algorithm and (2) virtual structure-based data collection algorithm.

### 2.2 Non-virtual structure-based data collection algorithm

A non-virtual structure-based data collection means that there is no auxiliary structure to assist the data collection, such as virtual grid, virtual honeycomb structure, and virtual ring structure. In this kind of algorithms, mobile sink either moves randomly or along a pre-determined trajectory. When mobile sink moves randomly and beyond the communication range of its previous neighbor sensor, a new neighbor sensor node of a mobile sink will be appointed as an agent node. These agent nodes can help the routing of sensory data. If the mobile sink moves along a pre-determined trajectory, the sensory data will always be routed to the sensor nodes near the trajectory.

In [15], Han et al. proposed the minimum Wiener index spanning tree (MWST), which is designed for IWSNs with a mobile sink. According to the characteristic of Wiener index, the MWST can provide efficient transmission paths for sensor nodes. However, finding a spanning tree with a minimal Wiener index from a weighted graph is a non-deterministic polynomial-time hardness (NP-hard) problem. Therefore, the authors proposed a new way to solve this problem; namely, through the extensive experiments, they found that the Wiener index of a minimum spanning tree (MST) is similar to the Wiener index of MWST and that time complexity of finding the MST is low. The authors used the Wiener index of MST as an initial upper bound. On this basis, the authors proposed two algorithms according to the network size. The first is a branch and bound algorithm for the small-scale sensor networks, and the second is a simulated annealing algorithm for the large-scale sensor networks. These algorithms provide a brand new idea for data transmission. However, the method to find the location of mobile sinks was ignored.

In [16], Shin and Kim proposed a milestone-based predictive routing protocol that can solve the problem of finding a spanning tree with the minimum Wiener index from a weighted graph presented in [15]. The proposed protocol consists of two main parts: estimation of mobile sink future location (namely, when a mobile sink finds some new sensor nodes entering its communication range, it broadcasts its updated location to them) and the establishment of milestone nodes and update of mobile

sink's location. The milestone nodes have to spread the estimated future location of a mobile sink to nodes located near the recent trail of a mobile sink. If the direction of a mobile sink is changed, it chooses a new neighbor sensor node as the next milestone node. All the sensory data are delivered by these milestone nodes. The milestone nodes are the tools of the source nodes to find the location of a mobile sink. However, too many milestone nodes are established if a mobile sink moves for a long time, which leads to a longer routing path. Consequently, the control packets among milestone nodes consume more energy. Thus, the presented way to find the mobile sink is inefficient.

A strategy of double cross to collect data for industrial wireless sensor networks was proposed by Shi et al. [17]. In this strategy, the authors introduced the double-blind concept where the source nodes and mobile sink do not know each other's location. The authors provided the scheme of a random line walk (RLW), which is used to transmit the sensory data. When a source node needs to upload the sensory data, it randomly chooses a direction to establish the baseline. The source nodes transmit sensory data in two directions in that baseline. On the basis of the RLW, the authors proposed the strategy of double cross. This strategy enables source nodes to be associated with a mobile sink. When a source node detects an event, it selects one direction randomly and transfers the sensory data along that direction and its vertical direction by the RLW mechanism. All sensor nodes on the transmission path cache the sensory data. When mobile sink needs sensory data, it transmits the query information. The query information is also transmitted in two directions, which are perpendicular to each other, according to the RLW mechanism. The query path and the sensory data routing path intersect at a sensor node. If the intersection sensor node receives both the query information and sensory data, the sensory data will be transmitted to mobile sink. In this strategy, too many sensor nodes need cache and relay sensory data to find the mobile sink; thus, energy is wasted, and transmission delay is large.

### 2.3 Virtual structure-based data collection algorithm

Data collection based on a virtual structure means that there is some auxiliary structure to assist the data gathering. The virtual structure contains the virtual grid, virtual honeycomb structure, virtual ring structure, and so on. The establishment of a virtual structure can simplify the location update of a mobile sink, data upload, path planning of a mobile sink, and so on. In this kind of algorithms, the movements of mobile sinks can be divided into three categories: random movement, fixed-trajectory movement, and dynamically adjusted trajectory movement.

#### 2.3.1 Random movement

Random movement of a mobile sink means that mobile sink can move in any direction at any speed during data collection.

In [18], Singh et al. proposed the EEGBDD algorithm. In this algorithm, every source node establishes its own virtual grid. Mobile sink moves randomly in the network, and sink initiates a query request when it needs data from the source node. The source node sends its data to the sink via the virtual grid. All query request and data are transferred through the dissemination nodes, and dissemination nodes are selected based on node residual energy and the distance from the node to the intersection of the grid. This algorithm reduces the length of the transmission path so that it reduces the energy consumption of nodes. However, some nodes may belong to more than one path so they could consume too much energy and die early.

Tunca et al. proposed an energy-efficient routing protocol to improve the network performance [19]. In this protocol, a ring structure is used to maintain of mobile sink location. After the network is deployed, a virtual ring structure consisted of nodes is built. Sink sends its location information to the nodes on the ring. When a node needs to send data, it first requests the location of a sink from the nodes on the ring and then sends data to the sink. To balance energy consumption, the nodes on the ring are periodically replaced.

In [20], Wang et al. proposed a grid-based data dissemination routing protocol. In this protocol, source node builds a grid to transmit data when an event is detected. Source node obtains the information on eight neighbor grid vertices near the source node and then sends a packet to all its neighbors. Every neighbor node replies a packet containing the information on node remaining energy. Source node selects the best neighbor as a relay node based on the residual energy of these neighbor nodes and distance from the sink to the neighbor nodes. The relay node repeats this process until the data arrive at the sink. When the mobile sink needs data, it initiates a query request. The request arrives at the source node through the relay nodes, and the source node sends data to the sink via the query path.

#### 2.3.2 Fixed-trajectory movement

When a mobile sink moves along a fixed trajectory in data collection, then, sink always moves along a pre-defined trajectory regardless the network status and application environment. Usually, the pre-defined trajectory contains the straight line, rectangle, cycle, and so on.

Mottaghi et al. proposed the O-LEACH algorithm [21] that represents an improved version of the LEACH algorithm. In the proposed algorithm, mobile sink moves

along a fixed line, whereas the area near the linear trajectory is called the convergence area. Nodes in the convergence area are set as convergence nodes (RNs). The network is divided into different clusters, and cluster head nodes are selected for each cluster. Data from the sensor nodes is sent to the cluster head nodes. Cluster head node transmits the data to the sink if it is near the sink; otherwise, cluster head node sends data to the near convergence node, and the convergence node transmits data to the sink. This algorithm reduces the energy of nodes, but a fixed trajectory of a mobile sink leads to the high energy consumption of nodes near the trajectory because these nodes bear more data forwarding task.

In [22], Konstantopoulos et al. proposed a data collection algorithm intended for an urban environment. In this algorithm, sink moves along a fixed trajectory and collects data from the nodes near the trajectory. A virtual structure called the cluster is established in the network. Cluster head node that is responsible for data fuse and data forwarding is selected in each cluster. Sensor nodes transmit data to their own cluster head nodes. Cluster head nodes send their data to those nodes near the trajectory of a mobile sink. This algorithm reduces the energy consumption of nodes. However, nodes near the trajectory of a mobile sink bear more task and expend their energy early.

Khan et al. proposed a data collect algorithm called the VGDR [23]. In this algorithm, a network is divided into virtual grids, and the number of the grids is only related to the number of nodes. Namely, each grid selects a head node, and the head node is in responsible for collecting data from nodes in its grid. Mobile sink moves along the network boundary, and during that movement, the routing of data is adjusted dynamically. Compared with the fixed line trajectory, the rectangle path of mobile sink balances energy consumption of the network but nodes near the trajectory also consume a lot of energy.

### 2.3.3 Dynamically adjusted trajectory movement

In this classification, a mobile sink is neither moving randomly nor moving on the pre-defined trajectory. However, the sink adjusts the trajectory dynamically according to the collected data, distribution of nodes, remaining energy of sensor nodes, and the number of travelling times in a region.

In [24], Kinalis et al. proposed an algorithm named the Biased Sink Mobility with Adaptive Stop Times. In this algorithm, the virtual grid structure is established in the network, and the intersection point of the grid is a stopping point of a mobile sink. If there is a higher density of nearby sensor nodes near the point, a mobile sink will stay longer at that point. The adaptive stop time of a mobile sink can balance the energy consumption of nodes, but a long trajectory of a mobile sink increases network delay.

Ghafoor et al. proposed the efficient trajectory design for mobile sink algorithm [25]. In this algorithm, the trajectory of a sink is based on a Hilbert curve, and trajectory of a mobile sink is adjusted dynamically according to the density of nodes and network size. The order of Hilbert curve is smaller when a mobile sink is moving toward the region with a smaller node density and vice versa. This algorithm can dynamically adjust the trajectory of mobile sink according to the network state so that it can balance energy consumption of nodes. However, this algorithm can only dynamically adjust the trajectory in a large range, but it is not suitable for data collection under the condition of uneven node density.

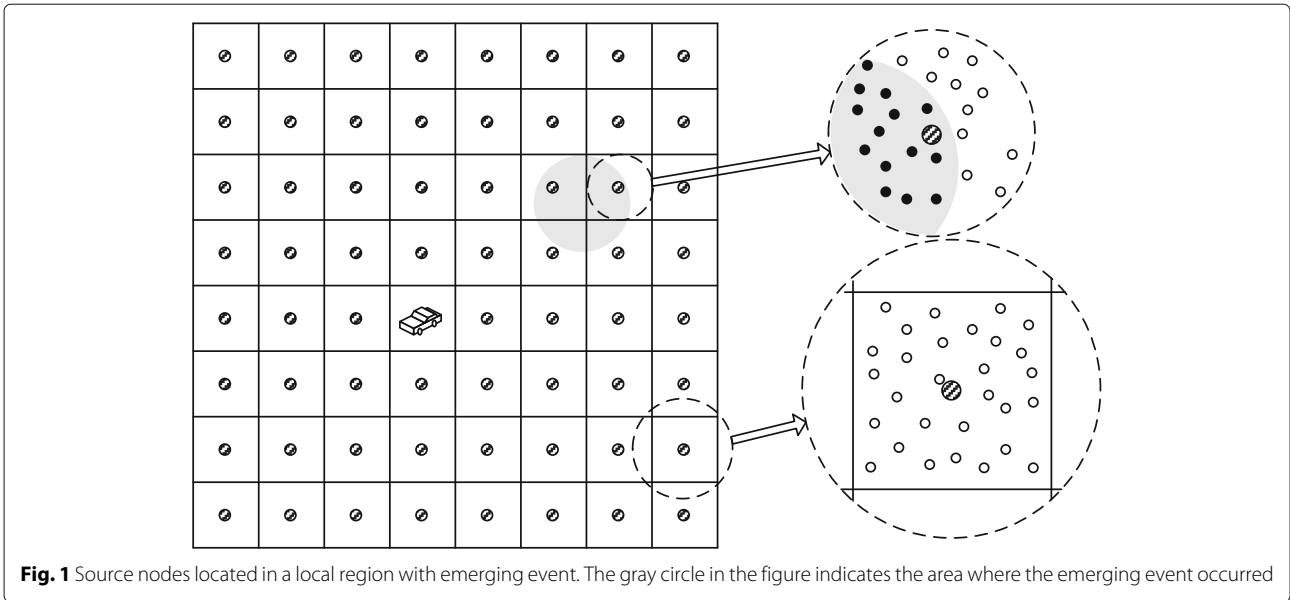
To solve the problem presented in [25], Yang et al. proposed the adjustable trajectory design based on node density for mobile sink algorithm [26]. In this algorithm, different orders of Hilbert curves are combined. By refining the established virtual grid structure and considering the density of nodes in a smaller region, Hilbert curves with different orders are constructed according to the different densities of nodes. Besides the fact that this algorithm solves the problem stated in [25], it also makes the data collection algorithm suitable for the case of uneven node deployment.

## 3 Method

### 3.1 Network model

Mobile data collection can prolong the network lifetime IWSNs. In most of mobile data collection algorithm, the application scenarios are always with evenly distributed events. However, in some real application scenarios, source nodes are distributed centrally in a local region, and the distributed region may change over the time. For instance, when the emergent events are monitored, such as monitoring of a fire in the industrial factory or monitoring the malfunction of industrial equipment, the source nodes are always distributed in a local region. As illustrated in Fig. 1, the sensor nodes within the event area are source nodes.

In the following, the network model and relevant assumptions are described in detail. The network is a rectangle area with the size of  $L \times D$ . The network consists of  $N$  sensor nodes, and these sensor nodes are deployed densely and randomly. All the sensor nodes are well connected, and the network is fully covered. The sensor nodes are static and location-aware (i.e., equipped with the GPS-capable antennas). Sensor nodes have the same initial energy, sensing radius  $r_s$ , and communication radius  $R$ . Sensory data is delivered to the mobile sink by a multi-hop transmission. There is only one mobile sink in the network and is not constrained by energy, memory space, and computing ability. The speed of the mobile sink  $v$  is fixed during in motion, and the mobile sink may park in



**Fig. 1** Source nodes located in a local region with emerging event. The gray circle in the figure indicates the area where the emerging event occurred

the center point of the virtual grid cell for a while. The sensor nodes are densely deployed, and there are no obstacles in the network, so we can assume that there are sensor nodes in each grid. The notation and the corresponding definition used in our algorithm are given in Table 1.

**3.2 Network initialization**

The first phase after the network is deployed is network initialization. This phase includes three sub-phases: establishment of a grid-based virtual structure, election of head nodes in the virtual grid cells, and establishment of the neighbor tables. Since the deployment area is a rectangle, we adopt the Cartesian coordinate system for convenience. The origin of this coordinate system is located at the lower left corner of the network.

**3.2.1 The establishment of grid-based virtual structure**

The establishment of a grid-based virtual structure includes three steps. The first step is to calculate the length of a virtual grid cell. The grid cell is a square in our algorithm, and calculation of a grid cell’s length is the foundation of a virtual structure. To set an ID number for each grid cell, we calculate the row-column number

(RCN) for each grid cell in step 2. To reduce the updating range of a mobile sink’s location, we introduce the direction number (DN) in step 3.

1) Calculation of side length of grid cells

As illustrated in Fig. 2, each grid cell has up to eight neighbor grid cells, and the sensor nodes in neighbor grid cells are called the neighbor nodes. To enable nodes to communicate with the neighboring nodes, the relation of communication radius  $R$  and a side length of grid cells  $a$  need to satisfy Eq. (1). When the side length  $a$  is calculated, mobile sink broadcasts some information in the network to establish the virtual structure. The information consists of  $a$ ,  $L$ , and  $D$ .

$$a \leq \frac{\sqrt{2}R}{4} \tag{1}$$

2) Calculation of RCN of grid cells

After receiving the information broadcasted by a mobile sink, sensor nodes calculate the number of grid cells in both horizontal direction and vertical direction. The calculating process is defined by Eqs. (2) and (3).

$$n_L = \left\lceil \frac{L}{a} \right\rceil \tag{2}$$

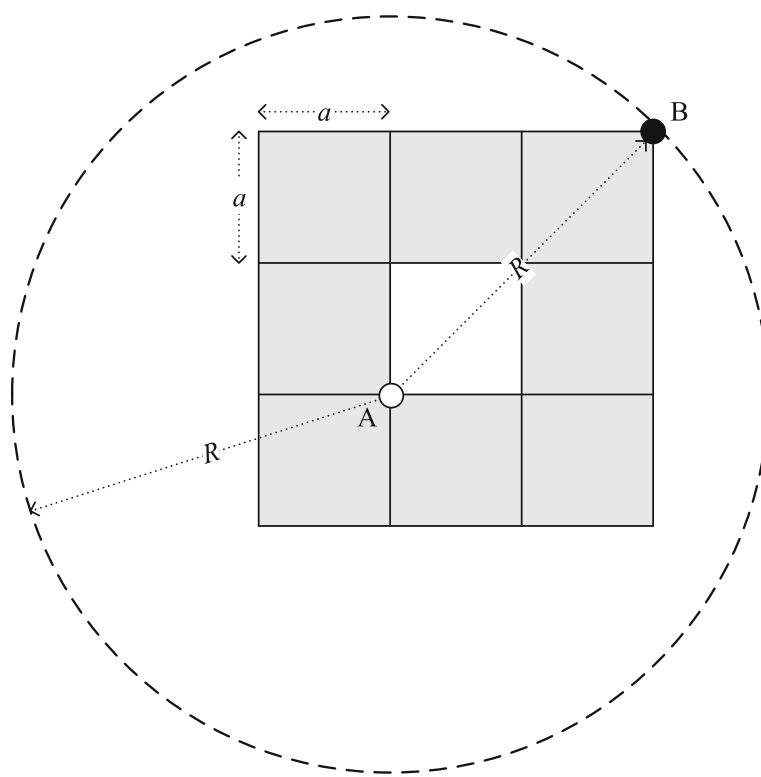
$$n_D = \left\lceil \frac{D}{a} \right\rceil \tag{3}$$

Based on the received information and their own coordinates, nodes can get an RCN, which indicates the location of a grid cell that the node belongs to; the calculating process is given in Eqs. (4) and (5).

$$M_r = \left\lceil \frac{y_i}{a} \right\rceil \tag{4}$$

**Table 1** Notations and definitions

Notation	Definition
$L$	Transverse length of network
$D$	Longitudinal length of network
$N$	Number of sensor nodes
$R$	Communication radius of sensor nodes
$r_s$	Sensing radius of sensor nodes
$v$	Speed of mobile sink



**Fig. 2** The relationship between communication radius and a side length of a grid cell. The distance between A and B is R, and it is easy to calculate the value of a

$$M_c = \left\lceil \frac{x_i}{a} \right\rceil \tag{5}$$

Sensor nodes in the same grid cell have the same RCN value, and the RCN will not be modified. When RCN is calculated, the network is as shown in Fig. 3.

3) Calculation DN of grid cells

Since the initial location of a mobile sink is in the grid cell center, sensor nodes can get the RCN of the grid cell center according to the information they received. The calculating process is shown in Eqs. (6) and (7).

$$M_{r\_sink} = \left\lceil \frac{nD}{2} \right\rceil = \left\lceil \frac{D}{2} \right\rceil \tag{6}$$

$$M_{c\_sink} = \left\lceil \frac{nL}{2} \right\rceil = \left\lceil \frac{L}{2} \right\rceil \tag{7}$$

After calculating the initial RCN of mobile sink, sensor nodes get the DN of grid cells that they belong to according to the relation of  $(M_{r\_sink}, M_{c\_sink})$  and their own RCN  $(M_r, M_c)$ . The comparing process is given in Eq. (8).

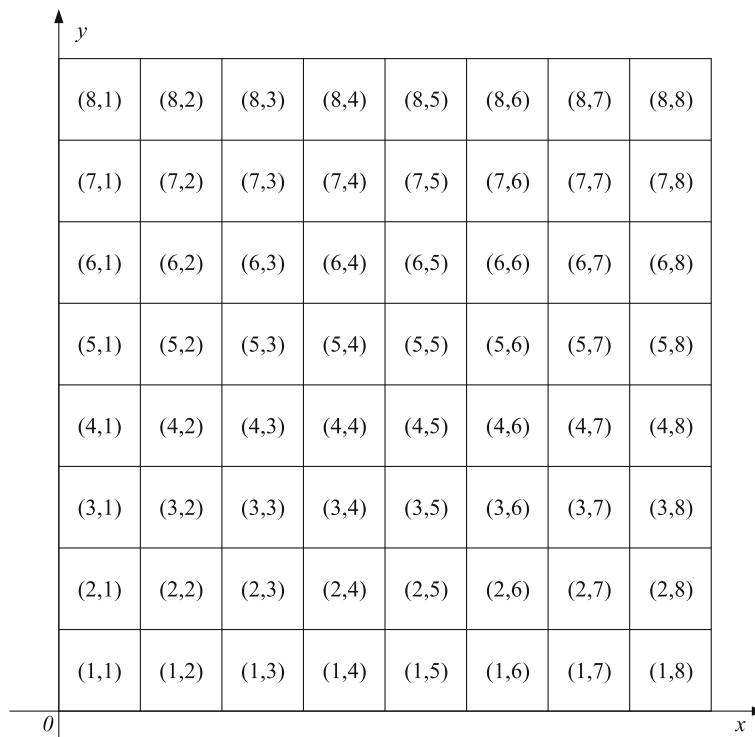
The grid cell where a mobile sink is located denoted the grid of sink (GS), and the relation between grid cell DN and GS is shown in Table 2 and Fig. 4.

$$DN = \begin{cases} 0, (M_r = M_{r\_sink}) \& (M_c = M_{c\_sink}) \\ 1, (M_r > M_{r\_sink}) \& (M_c > M_{c\_sink}) \\ 2, (M_r > M_{r\_sink}) \& (M_c < M_{c\_sink}) \\ 3, (M_r < M_{r\_sink}) \& (M_c < M_{c\_sink}) \\ 4, (M_r < M_{r\_sink}) \& (M_c > M_{c\_sink}) \\ 5, (M_r > M_{r\_sink}) \& (M_c = M_{c\_sink}) \\ 6, (M_r = M_{r\_sink}) \& (M_c < M_{c\_sink}) \\ 7, (M_r < M_{r\_sink}) \& (M_c = M_{c\_sink}) \\ 8, (M_r = M_{r\_sink}) \& (M_c > M_{c\_sink}) \end{cases} \tag{8}$$

3.2.2 The election of head nodes in virtual grid cells

In our VGDCA-C, the virtual grid cell head nodes are to collect sensory data from the same grid cell, deliver sensory data, and maintaining the DN of the grid cell.

In the first election, all the sensor nodes broadcast their coordinates and RCNs, and the radius of broadcasting is  $\sqrt{2}a$ . Thus, one sensor node can receive information from neighbor sensor nodes. Firstly, the sensor node compares its RCN with the RCN from the received information. If the received RCN is equal to that of the receiver, the coordinates and RCNs information will be added to the election list of the receiver. If the received RCN is not equal to that of the receiver, the receiver will drop the coordinates and RCN information. After the information is processed, each sensor node adds the source node of the information into its election list.



**Fig. 3** The RCN of virtual grid cell. It shows the RCS of virtual grid cell

After the operation of adding, sensor nodes sort their election lists. The election lists are sorted in ascending order by the abscissa, and if some sensor nodes have the same abscissas, the lists are sorted in ascending order by the ordinate. After the sorting operation, the sensor nodes in the same grid cell have the same election list. If a sensor node finds itself at the top of the election list, the sensor node broadcasts information of CellHead within the radius of  $\sqrt{2}a$ . The broadcasted CellHead consists of the indexes of sensor node in the election list. The other sensor nodes in the same grid cell receive the CellHead and

mark sending node as a head node of that cell. According to the sorted election list, the nodes in the grid are successively selected as the head nodes.

If some sensor nodes cannot broadcast CellHead, a waiting mechanism is introduced. If other nodes in the same grid cell cannot receive CellHead after a threshold period  $T_{time}$ , the next sensor node in the election list tries to broadcast CellHead.

**3.2.3 The establishment of neighbor tables**

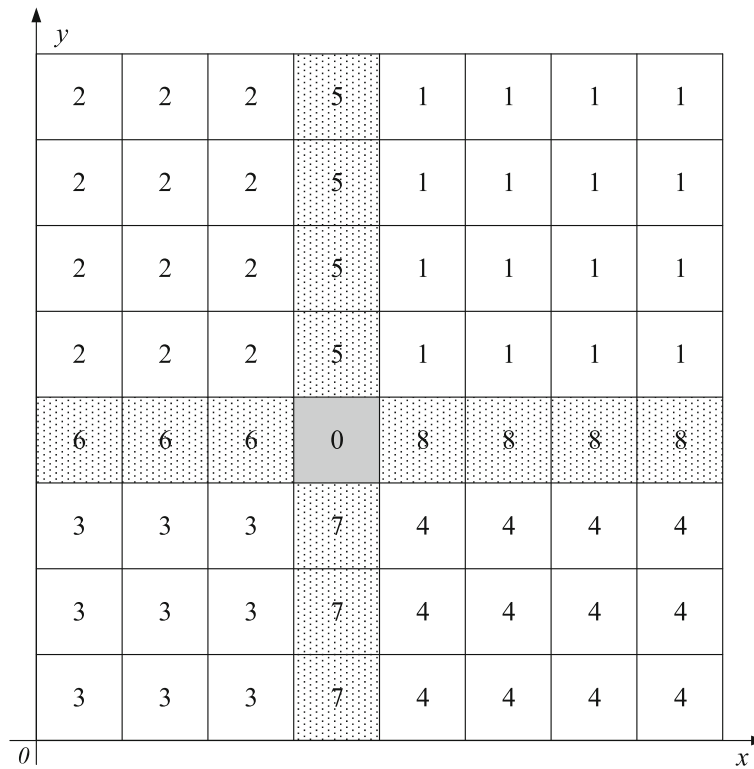
As each grid cell has up to eight neighbor grid cells, one head node has up to eight neighbor head nodes. In the process of establishing the neighbor tables, each head node broadcasts its coordinate and RCN in the communication range. Meanwhile, each head node can receive information from other head nodes. Each head node compares its RCN  $(M_r, M_c)$  with RCN of another head node  $(M'_r, M'_c)$ . According to the relation of  $(M_r, M_c)$  and  $(M'_r, M'_c)$ , neighbor head nodes can be added to the corresponding rows of the neighbor table. The neighbor table and corresponding criterion are shown in Table 3 and Fig. 5, respectively.

**Table 2** The relation between grid cells DN and GS

DN	Relation between corresponding grid cell and GS
0	The corresponding grid cell is GS
1	(up side of GS)&&(right side of GS)
2	(up side of GS)&&(left side of GS)
3	(down side of GS)&&(left side of GS)
4	(down side of GS)&&(right side of GS)
5	(same column with GS)&&(up side of GS)
6	(same row with GS)&&(left side of GS)
7	(same column with GS)&&(down side of GS)
8	(same row with GS)&&(right side of GS)

**3.3 Data routing**

After the network initialization, sensor nodes could rely on a virtual grid structure to upload sensory data to the



**Fig. 4** The DN of virtual grid cell. The grid cell 0 is the cell wherein a mobile sink is located

mobile sink. When source node has collected the sensory data, it encapsulates its RCN and sensory data into a data packet and transmits it to the corresponding head node in the same grid cell, and the head node forwards the data packet to a neighbor head node according to the DN of the former. The data packet also includes other information, and the detail will be described in the following sections. The relation of the DN of head nodes and the corresponding transmission direction are shown in Fig. 6 and Table 4.

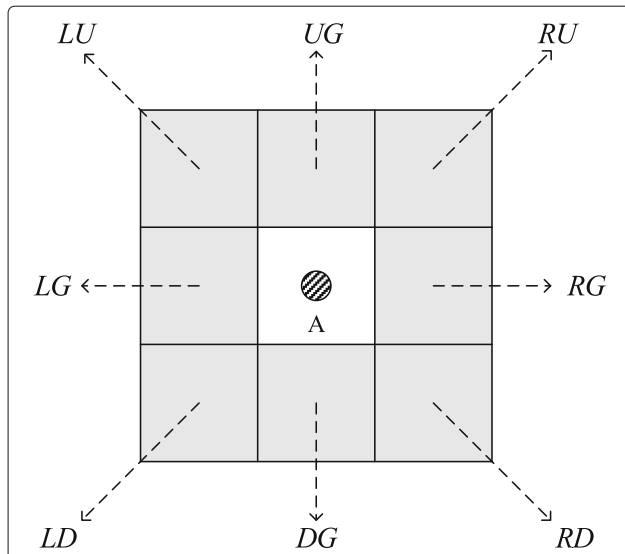
### 3.4 Trajectory planning of mobile sink

In our VGDCA-C, the moving region of a mobile sink is composed of a transverse moving belt and a longitudinal moving belt. The two moving belts are presented as a gray region in Fig. 7. The two moving belts may switch to another column or row when target events change. Before the switching of a moving belt, the mobile sink will complete Num<sub>cycle</sub> moving cycles in the moving belt. In Fig. 7, the grid cells in the gray region are called the moving belt grids. In one moving cycle, the mobile sink parks in

**Table 3** The neighbor table and corresponding criterion

Rows of neighbor table	Definition	Criterion
LU	Head node in the upper left corner	$(M'_r = M_r + 1) \& (M'_c = M_c - 1)$
RU	Head node in the upper right corner	$(M'_r = M_r + 1) \& (M'_c = M_c + 1)$
LD	Head node in the lower left corner	$(M'_r = M_r - 1) \& (M'_c = M_c - 1)$
RD	Head node in the lower right corner	$(M'_r = M_r - 1) \& (M'_c = M_c + 1)$
UG	Head node in the grid above	$(M'_r = M_r + 1) \& (M'_c = M_c)$
DG	Head node in the grid below	$(M'_r = M_r - 1) \& (M'_c = M_c)$
LG	Head node in the left grid	$(M'_r = M_r) \& (M'_c = M_c - 1)$
RG	Head node in the right grid	$(M'_r = M_r) \& (M'_c = M_c + 1)$





**Fig. 5** The neighbor grid cells and corresponding rows in neighbor table. Node A is the head node

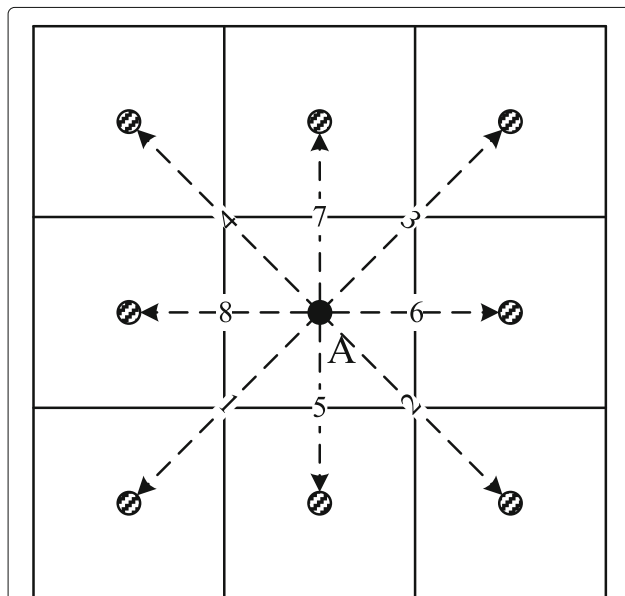
**Table 4** The rules of selecting next hop head node

DN of current head node	Transmission direction	Row selected table in neighbor
1	Bottom left	LD
2	Bottom right	RD
3	Top right	RU
4	Top left	LU
5	Down	DG
6	Right	RG
7	Up	UG
8	Left	LG
0	(Broadcast in local grid)	/

different moving belt grids for different durations according to the collected sensory data. The mobile sink assigns weight values to the moving belt grids, and the weight values affect the parking time of a mobile sink. When the mobile sink completes  $Num_{cycle}$  moving cycles in the current moving belts, it estimates the location of the event region. Subsequently, the moving belts switch according to the corresponding strategy.

**3.4.1 Trajectory of mobile sink in moving cycle**

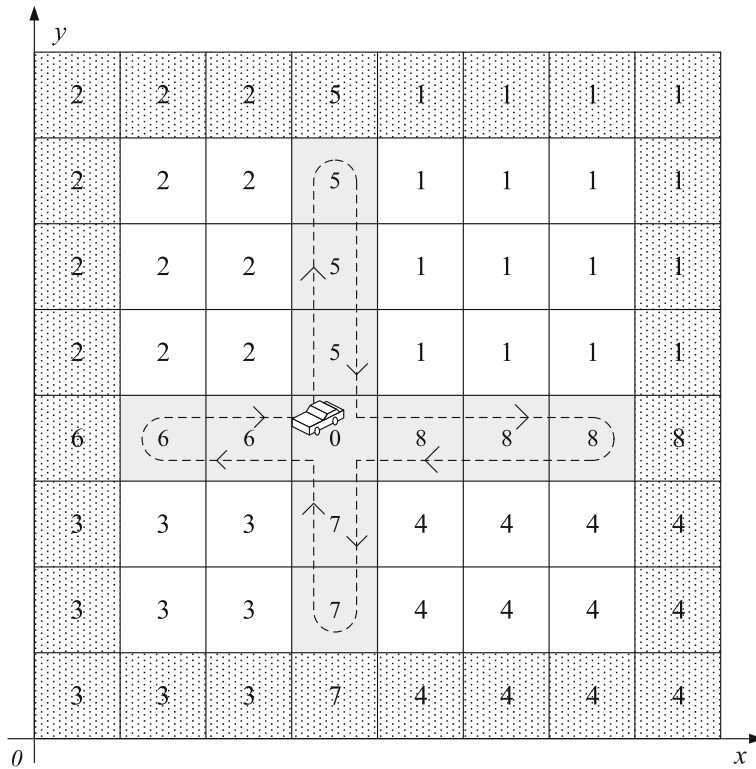
In our target application scenarios, the event region appears randomly. To adapt to this kind of application scenarios, we design a “cross” moving trajectory for a mobile sink. As illustrated in Fig. 7, the grid cell intersected by two moving belts is called the intersecting grid (IG). In one moving cycle, the mobile sink starts from IG and moves toward the direction whose DN is 5. After the mobile sink arrives at the grid cell whose  $M_r$  is  $(n_D - 1)$ , it starts to move in the opposite direction. The mobile sink can move back to the IG, and then, the moving direction switches to the direction whose DN is 8. When the mobile sink arrives at the grid cell whose  $M_c$  is  $(n_L - 1)$ , the mobile sink switches the moving direction to the opposite direction. The mobile sink moves in the direction whose DN is 7 after moving back to the IG. When the mobile sink arrives at the grid cell whose  $M_r$  is 2, it starts to move toward the IG again. After moving back to the IG, the mobile sink starts to move to the direction whose DN is 6 until it reaches the grid cell whose  $M_c$  is 2, then the mobile sink starts to move to the IG. When the mobile sink arrives at the IG, one moving cycle is completed.



**Fig. 6** The DN of current head node and corresponding transmission direction. Node A is the current head node, and the others are neighbor head nodes

**3.4.2 Calculation of weight value for moving belt grids**

According to Section 3.3, all the sensory data are routed to the moving belts. Different moving belt grids are responsible for different areas. As illustrated in Fig. 8, grid A has 13 responsible grid cells and grid B is responsible for 6 grid cells. Hence, different moving belts grids have different workloads. When a mobile sink starts the first moving cycle in the current moving belts, it calculates the weight value for each moving belt grid. Here,  $W$  is used to represent the weight value. The RCN of IG is  $(M_{r\_IG}, M_{c\_IG})$ , and the RCN of moving belt grid is  $(M_{r\_m}, M_{c\_m})$ . The process of calculation is as follows:



**Fig. 7** The moving region of mobile sink in a moving cycle. The dashed line with arrow is the trajectory of sink

$$(M_{r_m} > M_{r_{IG}}) \&\& (M_{c_m} = M_{c_{IG}}) :$$

$$W(M_{r_m}, M_{c_m}) = \min \left\{ (M_{c_m} - 1), \left( \left\lceil \frac{D}{a} \right\rceil - M_{r_m} \right) \right\} \\ + \min \left\{ \left( \left\lceil \frac{L}{a} \right\rceil - M_{c_m} \right) \left( \left\lceil \frac{D}{a} \right\rceil - M_{r_m} \right) \right\} \quad (9)$$

$$(M_{r_m} = M_{r_{IG}}) \&\& (M_{c_m} > M_{c_{IG}}) :$$

$$W(M_{r_m}, M_{c_m}) = \min \left\{ \left( \left\lceil \frac{D}{a} \right\rceil - M_{r_m} \right), \left( \left\lceil \frac{L}{a} \right\rceil - M_{c_m} \right) \right\} \\ + \min \left\{ \left( \left\lceil \frac{L}{a} \right\rceil - M_{c_m} \right), (M_{r_m} - 1) \right\} \quad (10)$$

$$(M_{r_m} < M_{r_{IG}}) \&\& (M_{c_m} = M_{c_{IG}}) :$$

$$W(M_{r_m}, M_{c_m}) = \min \left\{ \left( \left\lceil \frac{L}{a} \right\rceil - M_{c_{IG}} \right), (M_{r_m} - 1) \right\} \\ + \min \left\{ (M_{c_m} - 1), (M_{r_m} - 1) \right\} \quad (11)$$

$$(M_{r_m} = M_{r_{IG}}) \&\& (M_{c_m} < M_{c_{IG}}) :$$

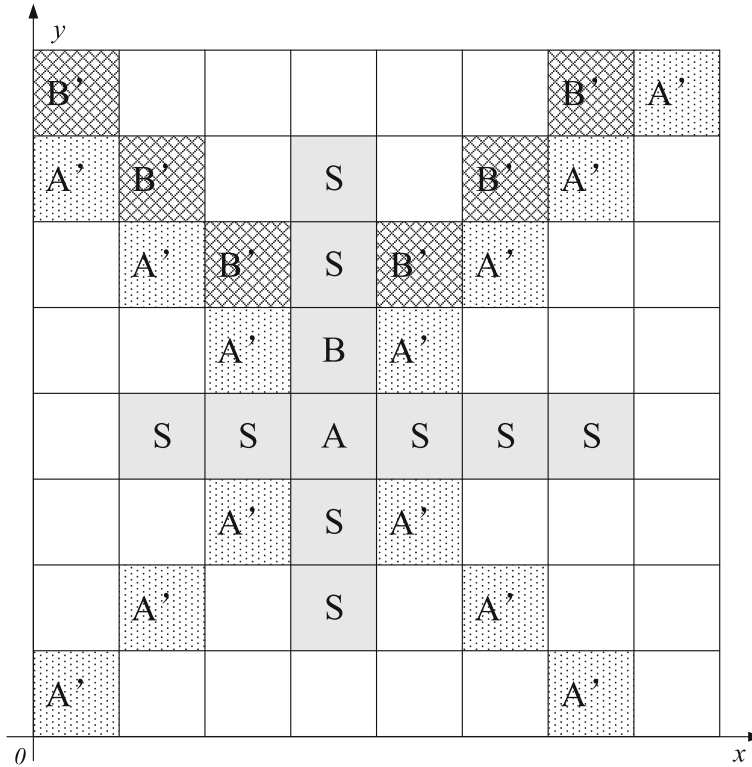
$$W(M_{r_m}, M_{c_m}) = \min \left\{ \left( \left\lceil \frac{D}{a} \right\rceil - M_{r_m} \right), (M_{c_m} - 1) \right\} \\ + \min \left\{ (M_{r_m} - 1), (M_{c_m} - 1) \right\} \quad (12)$$

$$(M_{r_m} > M_{r_{IG}}) \&\& (M_{c_m} = M_{c_{IG}}) :$$

$$W(M_{r_m}, M_{c_m}) = \min \left\{ (M_{c_{IG}} - 1), \left( \left\lceil \frac{D}{a} \right\rceil - M_{r_{IG}} \right) \right\} \\ + \min \left\{ \left( \left\lceil \frac{D}{a} \right\rceil - M_{r_{IG}} \right), \left( \left\lceil \frac{L}{a} \right\rceil - M_{c_{IG}} \right) \right\} \\ + \min \left\{ \left( \left\lceil \frac{L}{a} \right\rceil - M_{c_{IG}} \right), (M_{r_{IG}} - 1) \right\} \\ + \min \left\{ (M_{r_{IG}} - 1), (M_{c_{IG}} - 1) \right\} \quad (13)$$

### 3.4.3 Allocation of parking time for mobile sink in the first moving cycle

When the mobile sink starts the first moving cycle in the current moving belt, it allocates parking time for the moving belt grids according to the corresponding weight values. In one moving cycle, the moving time of a mobile



**Fig. 8** The moving belt grids and corresponding responsible areas. *S*, *A*, and *B* are the moving regions of mobile sink in one moving cycle. *A'* represents the region which grid *A* is responsible for and *B'* represents the region which grid *B* is responsible for

sink is calculated by Eq. (14). We assume that in a moving cycle, the parking time  $t_{pa}$  is equal to the moving time  $t_{moving\_sink}$ . Therefore, the total time of a moving cycle is calculated by Eq. (15).

$$t_{moving\_sink} = \frac{2 \times a \times (n_L + n_D - 6)}{v} \tag{14}$$

$$T = t_{moving\_sink} + t_{pa} = \frac{4 \times a \times (n_L + n_D - 6)}{v} \tag{15}$$

We allocate  $t_{pa}$  to each moving belt grid according to the corresponding weight values, and the allocated time is the corresponding parking time. Thus, we need to get the sum of weight values, which is defined in Eq. (16).

$$W_{all} = \sum_{i=2}^{n_D-1} W(i, M_{c\_IG}) + \sum_{j=2}^{n_L-1} W(M_{r\_IG}, j) - W(M_{r\_IG}, M_{c\_IG}) \tag{16}$$

In the allocation process, the moving belt grids need to be classified into three categories. The first category is a top grid, and it refers to the moving belt grids that are adjacent to the boundary grids. The second category is IG, and the third category consists of other moving belt grids. The reason for grids classification is that the num-

ber of parking in different types of grid cells is different in a moving cycle. If a grid cell is a top grid, a mobile sink can park in this grid cell only once. We assume that the RCN of a moving belt grid is  $(i, j)$ . The number of times the mobile sink parks in different types of moving belt grids is calculated as follows:

- If a grid  $(i, j)$  is a top grid, then:

$$t_p = \frac{W(i, j)}{W_{all}} \times t_{pa} \tag{17}$$

- If a grid  $(i, j)$  is IG, then:

$$t_p(i, j) = \frac{W(i, j)}{4} \times t_{pa} \tag{18}$$

- If a grid  $(i, j)$  is the other moving belt grids, then

$$t_p(i, j) = \frac{W(i, j)}{2} \times t_{pa} \tag{19}$$

In the first moving cycle, a mobile sink starts to collect sensory data in the current moving belt according to the

parking time given above. When a mobile sink collects the sensory data, it counts the sensory data in the corresponding counters. Therefore, in the following, we will talk about the counters in the mobile sink.

### 3.4.4 Counters of mobile sink

In the process of sensory data routing to the mobile sink, the sensory data carry the information on source nodes and routing. According to that information, the mobile sink counts the corresponding counters. When the sensory data are sensed, the source nodes attach their RCNs to the sensory data. When the sensory data are routed to the moving belt grid for the first time, then RCN of this moving belt grid is attached to the sensory data. The counters the mobile sink has are as follows:

- $C_{gm}(i, j)$ : In the sensory data, if the RCN of a moving belt grid that is first met is  $(i, j)$ , then the amount of sensory data is added to this counter.
- $C_{gm}(i, j)$ : This counter is used to record the number of the moving belt grids which the sensory data pass and the amount of sensory data are added to the counters corresponding to these passed moving belt grids.
- $C_{horizontal}(i)$ : The value of  $i$  is from 1 to 3. As the RCN of the source node is in the sensory data,  $M_r$  of the source node can be obtained. If  $M_r$  of the source node is larger than IGs, the value of  $i$  is 1, and the amount of sensory data is added to  $C_{horizontal}(1)$ . If  $M_r$  of the source node is equal to IGs, the value of  $i$  is 2. The value of  $i$  is 3 when the  $M_r$  of the source node is smaller than IGs.
- $C_{vertical}(i)$ : The value of  $i$  is from 1 to 3. As the RCN of the source node is in the sensory data,  $M_c$  of the source node can be obtained. If  $M_c$  of the source node is larger than IGs, the value of  $i$  is 1 and the amount of sensory data is added to  $C_{vertical}(1)$ . If  $M_c$  of the source node is equal to IGs, the value of  $i$  is 2. The value of  $i$  is 3 when the  $M_c$  of the source node is smaller than IGs.

### 3.4.5 Dynamic adjustment of parking time

After completing the first moving cycle and moving back to the IG, a mobile sink readjusts the parking time for the next moving cycle according to its counters. Due to that, different grid cells handle different amounts of sensory data in the moving belts, so the mobile sink needs to park longer in the moving belt grid, which handles more sensory data. We divide the time of a moving cycle into two parts on average, of which is allocated by  $C_{gm}(i, j)$ , and another is allocated by  $C_{gr}(i, j)$ . The summing process of  $C_{gm}(i, j)$  is defined by Eq. (20), and the summing process of  $C_{gr}(i, j)$  is defined by Eq. (21).

$$C_{gma} = \sum_{j=2}^{n_L-1} C_{gm}(M_{r\_IG}, j) + \sum_{i=2}^{n_D-1} C_{gm}(i, M_{c\_IG}) - C_{gm}(M_{r\_IG}, M_{c\_IG}) \quad (20)$$

$$C_{gra} = \sum_{j=2}^{n_L-1} C_{gr}(M_{r\_IG}, j) + \sum_{i=2}^{n_D-1} C_{gr}(i, M_{c\_IG}) - C_{gr}(M_{r\_IG}, M_{c\_IG}) \quad (21)$$

When the recalculated parking time is discussed, the moving belt grids need to be classified into three categories: top grids, IG, and other moving belt grids. We assume that the RCN of a moving belt grid is  $(i, j)$ . The time that mobile sink parking in different types of moving belt grids is calculated as follows (the calculated time is the time of one parking in the grid):

- When a grid  $(i, j)$  is a top grid the time is defined by

$$t_p(i, j) = \frac{C_{gm}(i, j)}{C_{gma}} \times \frac{t_{pa}}{2} + \frac{C_{gr}(i, j)}{C_{gra}} \times \frac{t_{pa}}{2} = \left[ \frac{C_{gm}(i, j)}{C_{gma}} + \frac{C_{gr}(i, j)}{C_{gra}} \right] \times \frac{a \times (n_L + n_D - 6)}{v} \quad (22)$$

- When a grid  $(i, j)$  is IG the time is defined by

$$t_p(i, j) = \frac{C_{gm}(i, j)}{C_{gma}} \times \frac{t_{pa}}{2} + \frac{C_{gr}(i, j)}{C_{gra}} \times \frac{t_{pa}}{2} = \left[ \frac{C_{gm}(i, j)}{C_{gma}} + \frac{C_{gr}(i, j)}{C_{gra}} \right] \times \frac{a \times (n_L + n_D - 6)}{4 \times v} \quad (23)$$

- When a grid  $(i, j)$  is the other moving belt grid the time is defined by

$$t_p(i, j) = \frac{C_{gm}(i, j)}{C_{gma}} \times \frac{t_{pa}}{2} + \frac{C_{gr}(i, j)}{C_{gra}} \times \frac{t_{pa}}{2} = \left[ \frac{C_{gm}(i, j)}{C_{gma}} + \frac{C_{gr}(i, j)}{C_{gra}} \right] \times \frac{a \times (n_L + n_D - 6)}{2 \times v} \quad (24)$$

After the redistribution of parking time, a mobile sink starts a new moving cycle according to the allocated

parking. Before the start of a new moving cycle, all the counters of a mobile sink are set to zero. After the mobile sink completes  $\text{Num}_{\text{cycle}}$  moving cycles, it estimates the location of the event area according to the counters. The moving belt then moves toward the event area.

### 3.4.6 Moving trajectory of moving belts

As the source nodes are distributed centrally, the mobile sink needs to move toward the event area as close as possible such that the total length of the sensory data transmission is reduced and the energy consumption is decreased. Before the moving belts switch to another column or row, the mobile sink first compares  $C_{\text{horizontal}}(1)$ ,  $C_{\text{horizontal}}(2)$ , and  $C_{\text{horizontal}}(3)$ . If  $C_{\text{horizontal}}(1)$  has the largest value, the transverse moving belt switches to a neighbor row above. If  $C_{\text{horizontal}}(3)$  has the largest value, the transverse moving belt switches to a neighbor row below. If  $C_{\text{horizontal}}(2)$  is the largest, or two of three counters are equal, or all three counters are equal, the transverse moving belt does not switch to another row. Similarly, the mobile sink compares  $C_{\text{vertical}}(1)$ ,  $C_{\text{vertical}}(2)$  and  $C_{\text{vertical}}(3)$ . If  $C_{\text{vertical}}(1)$  has the largest value, the longitudinal moving belt switches to a neighbor column on the right. If  $C_{\text{vertical}}(3)$  is the largest, the longitudinal moving belt switches to a neighbor column on the left. In other cases, the longitudinal moving belt does not switch. According to this strategy, the mobile sink can get closer to the event area.

### 3.5 Local updating of mobile sink

When a mobile sink moves from one grid cell to another, the DN of grid cells needs to be updated to maintain a reliable data routing. According to Section 3.3, the sensory data are routed to the mobile sink according to DN of grid cells. Hence, the updating of DN of grid cells denotes the updating of the location information of a mobile sink. For further discussion, the updating operations need to be classified into two categories.

The first category refers to the situation that the moving belts do not switch. In this situation, DNs of two grid cells need to be updated when the mobile sink enter into a new grid cell. The first grid cell is the one that the mobile sink is located in before entering the new one. The second grid cell is the one that the mobile sink entered, and this grid cell sets its DN to 0. The former grid cell sets its DN according to the moving direction. If mobile sink moves toward the direction whose DN is 5, the DN of the former grid cell is set to 7. If the direction DN is 8, the former's DN is set to 6. If the direction has DN is 7, the former's DN is set to 5. If the direction DN is 6, the former's DN is set to 8.

The second category refers to the situation that the moving belts switch to another column or row. When a transverse moving belt switches to another row, both previous row and current row need to update their DNs. When a longitudinal moving belt switches to another column, both previous column and current column need to update their DNs. The updating is relevant to the grid cell where the mobile sink is located before moving. The relationship of grid cell DN and GS is shown in Fig. 4. Namely, when mobile sink updates a grid cell's DN, it only needs to update the head node of the grid cell. Thus, the local updating not only reduces the update area but also reduces the number of the updated sensor nodes.

### 3.6 Re-election of head nodes

To prolong the network lifetime and balance the energy consumption of sensor nodes in each grid cell, the head nodes need to be regularly re-elected. First, the ratio of the residual energy of the current head node and its energy when it was elected is calculated. If that ratio is below a threshold  $\text{Th}$ , the head node broadcasts *Re-Election* to start re-election of the corresponding grid cell. The *Re-Election* includes the ordinal number of the head nodes in the election list *Index*. The sensor nodes in the same grid cell can receive the Re-Election and query the sensor node whose the ordinal number in the election list is  $(\text{Index} + 1)$ . If a sensor node *A* finds that the sensor node queried itself, the sensor node *A* sends *CellHead\_01* to the current head node. After receiving *CellHead\_01*, the current head node sends its neighbor list to the sensor node *A*. When sensor node *A* receives the neighbor list of the current head node, it broadcasts *CellHead\_02*. When the other sensor nodes receive *CellHead\_02*, the sensor node *A* becomes the new head node of the current grid cell, and the previous head node is retired. According to Eq. (1), all eight neighbor head nodes can receive *CellHead\_02*, and the neighbor head nodes then add sensor node *A* to their neighbor lists. Before the new head node is ready, the sensory data are also transmitted to the previous head node. When the other sensor nodes are waiting for *CellHead\_02*, if the waiting time exceeds the threshold  $\text{Th}_{\text{time}}$ , they query the sensor node whose the ordinal number is  $(\text{Index} + 2)$ . If the current head node is the last one on the election list, then the sensor nodes will query the first sensor node on the election list in the next election.

## 4 Results and discussion

### 4.1 Simulation model

To evaluate the performance of our proposed algorithm, we designed the simulation experiment using the MATLAB platform. The simulation parameters are listed in Table 5 and the values of the parameters are listed in Table 6. In our simulation, the target monitoring area was a rectangle area with the size of  $L \times D$ . The network

**Table 5** Simulation parameters

Parameters	Definition
$L$	Transverse length of network
$D$	Longitudinal length of network
$N$	Number of sensor nodes
$r_c$	Communication radius of sensor nodes
$r_e$	Radius of event area
$E$	Initial energy of sensor node
$d_0$	Threshold of communication
$l_c$	Size of control packet
$l_s$	Size of sensory data packet
$E_{elec}$	Digital electronics
$E_{fs}$	Communication ( $d < d_0$ )
$E_{mp}$	Communication ( $d \geq d_0$ )
Th	Threshold of re-election
Num <sub>cycle</sub>	Threshold of movig cycle
$v_{sink}$	Velocity of mobile sink

consisted of  $N$  sensor nodes, and these sensor nodes had the same communication radius  $r_c$  and initial energy  $E$ . The size of the control pocket was  $l_c$ , and the size of the sensory data packet was  $l_s$ . We adopted the energy model wherein if a sensor node receives  $n_b$  bit sensory data, the consumed energy is  $(n_b \times E_{elec})J$ ; if a sensor node sends  $n_b$  bit sensory data, the consumed energy can be classified into two categories. We assumed the transmission distance is  $d$ , and the threshold of transmission distance is  $d_0$ . If  $d < d_0$ , the consumed energy is  $[n_b \times (E_{elec} + E_{fs} \times d^2)]J$ ; if  $d \geq d_0$ , the energy is equal to  $[n_b \times (E_{elec} + E_{mp} \times d^4)]J$ .

**Table 6** Values of simulation parameters

Parameters	Values
$L$	200 m
$D$	200 m
$N$	800~1600
$r_c$	75 m
$r_e$	30 m
$E$	2 J
$d_0$	75 m
$l_c$	200 bit
$l_s$	800 bit
$E_{elec}$	50 nJ/bit
$E_{fs}$	10 pJ/bit/m <sup>2</sup>
$E_{mp}$	0.0013 pJ/bit/m <sup>4</sup>
Th	10~90%
Num <sub>cycle</sub>	2, 4, 6, 8, 10
$v_{sink}$	10 m/s

To simulate the centrally distributed event, we assumed event area is a circular area and all sensor nodes in this area are appointed as source nodes. The network lifetime was determined by the time when the first node runs out of its energy. In our VGDC-A-C, the real-time data collection means that the source nodes upload sensory data immediately when the sensory data is sensed. In this context, we mainly focused on the balance of energy consumption to prolong the network lifetime. Thus, we focused on the lifetime performance and energy consumption balance, while the delay of sensory data transmission was not considered. We used the following aspects as performance metrics of simulation experiments:

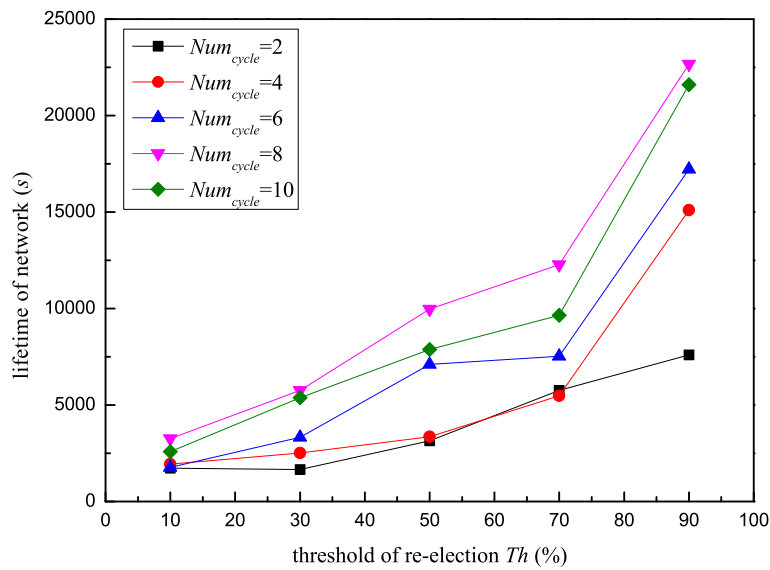
- Network lifetime: defined by the time when the first node runs out of its energy
- Average residual energy: the mean value of residual energies of all the sensor nodes
- A variance of residual energy: the variance of residual energies of all the sensor nodes
- An average number of transmission hops: the mean value of transmission hops from a source node to the mobile sink.

## 4.2 Performance analysis under different system parameters

In the simulation experiments of VGDC-A-C, we first studied the impact of system parameters Th and Num<sub>cycle</sub> on network performances, where Th denotes the threshold of the re-election of head nodes, which affects the equilibrium of energy consumption in a single grid cell, and Num<sub>cycle</sub> affects the energy consumption in the entire network and network lifetime. In these experiments, the number of sensor nodes was 1000, and the location of event area changed every 1000 s. The values of Th were 10, 30, 50, 70, and 90%, and the values of Num<sub>cycle</sub> were 2, 4, 6, 8, and 10.

### 4.2.1 Network lifetime

As illustrated in Fig. 9, the network lifetime was minimal at Th of 10%, and the lifetime increased with the increase of Th. In the case of Th of 10%, the head node could start re-election when the residual energy became 10% of the energy at the election time. It means that sensor nodes that had been head nodes had little residual energy. However, the sensor nodes that had not been elected had higher residual energy. Thus, the energy consumption within the grid cell was unbalanced, and the first sensor node that ran out of energy appeared soon. Therefore, the network lifetime was short when Th was 10%. With the increase of Th, the sensor nodes in the grid cell assumed the task of the head node more frequently, and the energy consumption within grid cell became more and



**Fig. 9** The network lifetime under different system parameters ( $Th$  and  $Num_{cycle}$ ). When  $Th$  was 90% and  $Num_{cycle}$  was 8, the network lifetime could reach the maximum value

more balanced. We found that the network lifetime had the highest growth rate when  $Th$  varied between 70 and 90% because in that case, energy consumption within grid cell was the most balanced.

When we analyzed the situation from the vertical axis, we found that for the same  $Th$ , network lifetime was minimal when  $Num_{cycle}$  was 2. When  $Num_{cycle}$  was 2, the mobile sink needed to determine whether to switch the moving belts after every two completed moving cycles. According to the distribution of event area and the strategy of switching the moving belts, two moving belts switched in general circumstances. Hence, the grid cells of two rows and two columns needed to be updated. Therefore, much energy was consumed and network lifetime was shortened. With the increase of  $Num_{cycle}$ , the network lifetime also increased. When  $Num_{cycle}$  reached 8, the network lifetime had the maximum value because the updating operation did not consume much energy when the moving belts did not switch frequently. When  $Num_{cycle}$  was 10, the network lifetime began to decrease because all the sensory data were routed to the moving belts. If the moving belts were not switched for a long time, the sensor nodes in the moving belts could cause the “hot spot” problems, which would shorten the network lifetime. Thus, when  $Th$  was 90% and  $Num_{cycle}$  was 8, the network lifetime could reach the maximum value.

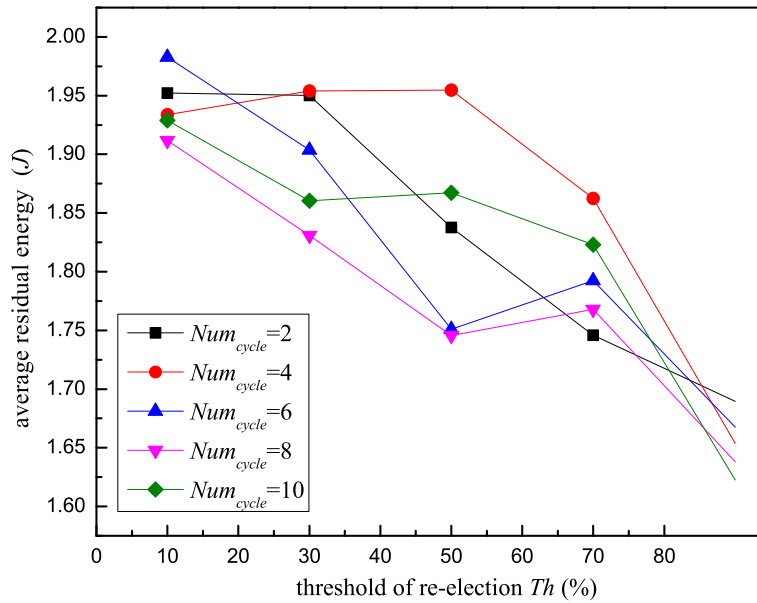
**4.2.2 Average residual energy**

The average residual energy indicates the energy utilization rate of the network. As illustrated in Fig. 10, with the increase of  $Th$ , the average residual energy decreased continuously. According to Fig. 9, the network lifetime

increase was caused by the increase of  $Th$ . Thus, the average residual energy decreased. When  $Th$  was unchanged, the average residual energy was the lowest at  $Num_{cycle}$  of 8, which means that the energy utilization rate was the best in that case. In Fig. 10, it can be seen that when  $Num_{cycle}$  was equal to 2, the average residual energy was lower than the average residual energy at  $Num_{cycle}$  of 4. However, when  $Num_{cycle}$  was 2, the network lifetime was shorter than that when  $Num_{cycle}$  was 4. This is because the moving belts needed to be constantly switched, and much energy was consumed to update the corresponding grid cells. Thus, the energy utilization rate was not high when  $Num_{cycle}$  was 2 because much energy was wasted. On the other hand, when  $Num_{cycle}$  was 8, the energy utilization rate was the highest.

**4.2.3 Variance of residual energy**

The variance of residual energy indicates the balance of energy consumption. As illustrated in Fig. 11, the variance was minimal at  $Th$  of 10% because only a small number of sensor nodes had consumed energy when the network ended. Most sensor nodes had almost the same residual energy. With the increase of  $Th$ , most sensor nodes began to consume energy, and the variance increased. When the  $Num_{cycle}$  was 2 and 4, the network lifetime was short, and the variance was low. Although the energy consumption was balanced in that state, that was not an excellent situation. When the  $Num_{cycle}$  was 10, the variance reached the maximum value because in that case, the moving belts stayed in the same area for a long time, and the sensory data converged in that area. Hence, the sensor nodes of that area consumed all energy very fast. The energy



**Fig. 10** The average residual energy under different  $Th$  and  $Num_{cycle}$ . When  $Num_{cycle}$  was 8, the energy utilization rate was the highest

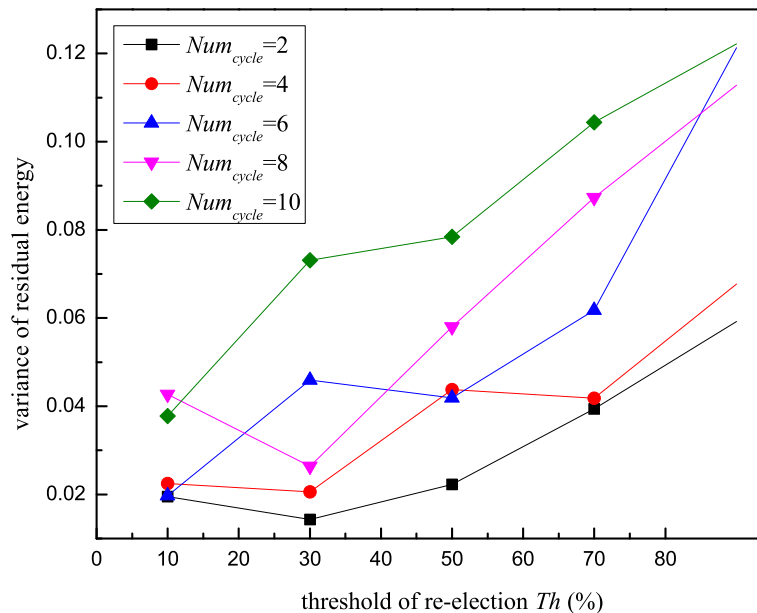
consumption of network was unbalanced. When  $Num_{cycle}$  was 6 and 8, the variance values were close, and the network had a better performance regarding the lifetime and average residual energy.

According to the analysis of network lifetime, average residual energy, and residual energy’s variance, the network had the best performance at  $Th$  of 90% and

$Num_{cycle}$  of 8. There we set  $Th$  to 90% and  $Num_{cycle}$  to 8 and compared the performance of the VGDC-A-C and VGDD [27].

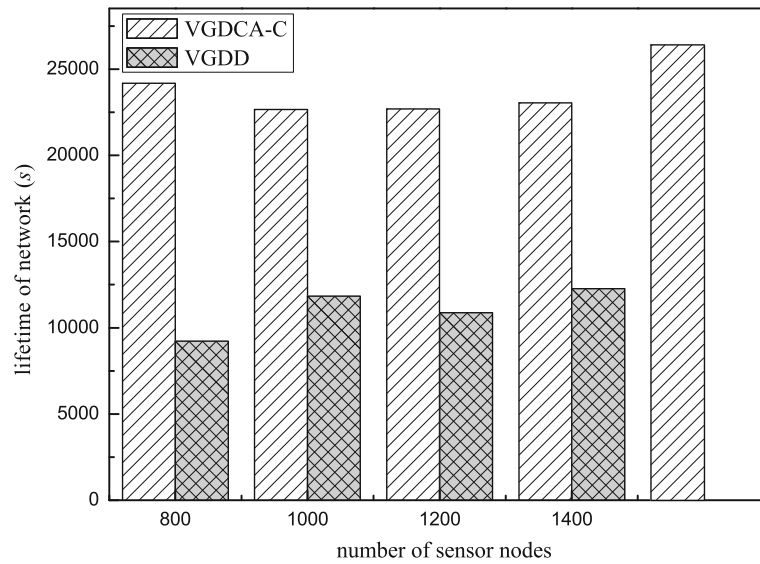
### 4.3 Comparison with VGDD

In this section, we compare the performance of the VGDC-A-C and VGDD for different numbers of sensor



**Fig. 11** The variance of residual energy under different system parameters ( $Th$  and  $Num_{cycle}$ ). When  $Num_{cycle}$  was 6 and 8, the variance values were close, and the network had a better performance regarding the lifetime and average residual energy





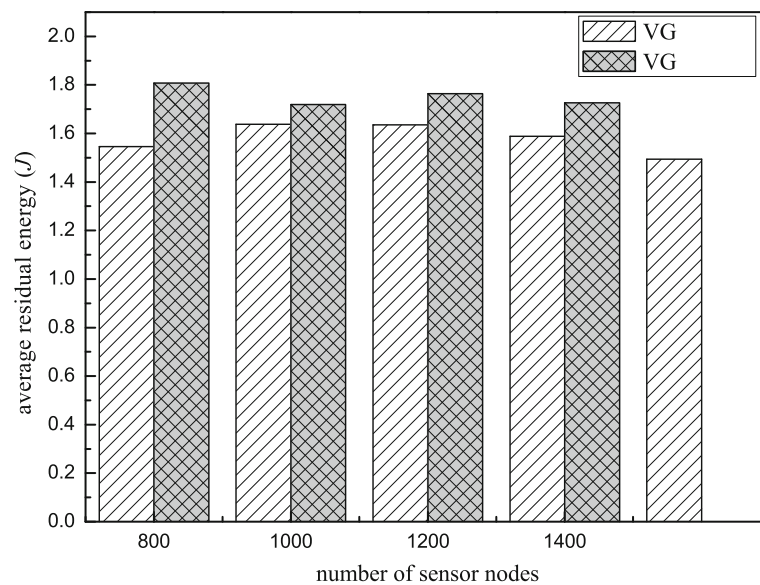
**Fig. 12** The comparison of VGDD and VGDC-A-C in network lifetime. It shows that network lifetime of the VGDC-A-C was longer than VGDD in our simulation model

nodes. The numbers of sensor nodes were 800, 1000, 1200, 1400, and 1600, respectively, and the time interval of event area change was 1000 s.

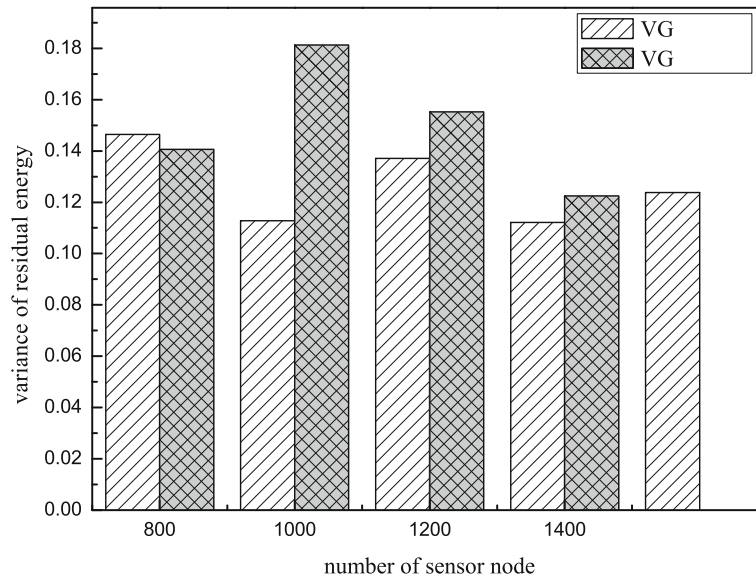
**4.3.1 Network lifetime**

As illustrated in Fig. 12, we compared the network lifetime for different numbers of sensor nodes. With the increase of the number of sensor nodes, the network lifetime of the VGDC-A-C was always larger than that of the VGDDs. In the application scenarios with centrally

distributed events, the VGDC-A-C could allocate parking time dynamically according to the counters of a mobile sink. The mobile sink would park longer time in the virtual grids with more sensory data. This method could reduce energy consumption and a total length of transmission. Meanwhile, two moving belts would switch dynamically according to the counters of a mobile sink. These counters indicated the location of event area, and the moving belts switched toward the event area. However, in the VGDD, a mobile sink moved by the predefined trajectory. When the



**Fig. 13** The comparison of VGDD and VGDC-A-C in average residual energy. It indicates that the VGDC-A-C could work longer when the same energy was consumed, which further means that the VGDC-A-C had higher energy utilization ratio

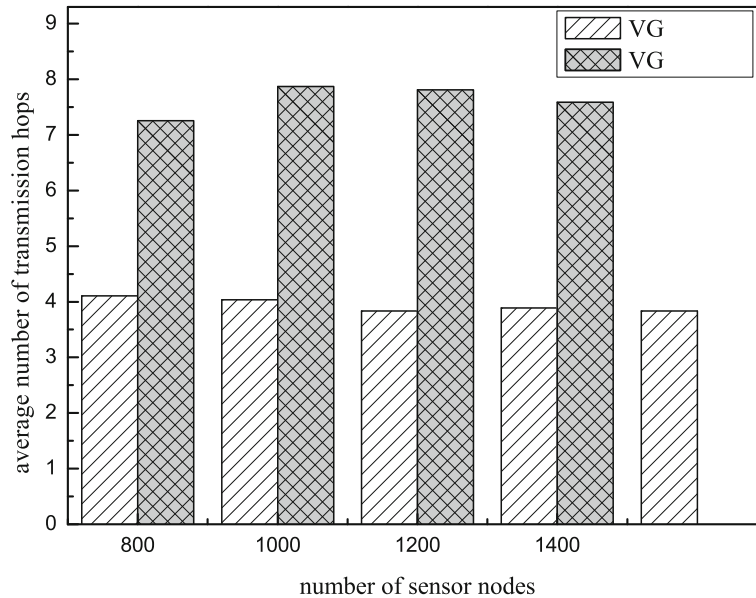


**Fig. 14** The comparison of VGDD and VGCA-C in a variance of residual energy. The energy consumption balance of the VGCA-C was slightly better than that of the VGDD

event area changed, the mobile sink of the VGDD could not adjust its trajectory to the event area. Hence, network lifetime of the VGCA-C was longer. When the number of sensor nodes increased, the number of source nodes also increased. Moreover, there was no major fluctuation in the network lifetime due to the increase of the number of sensor nodes.

#### 4.3.2 Average residual energy

As mentioned above, the average residual energy indicates the energy utilization ratio. As illustrated in Fig. 13, with the increase of the number of sensor nodes, the average residual energy of the VGCA-C was slightly lower than that of the VGDD, which indicates that energy utilization ratio of the VGCA-C was higher than that of the



**Fig. 15** The comparison of VGDD and VGCA-C in an average number of transmission hops. The VGCA-C decreases transmission hops in the applications with centrally distributed events

VGDD. In Fig. 12, it can be seen that network lifetime of the VGDC-A-C was two times greater than that of the VGDD. However, the average residual energy of VGDC-A-C was only slightly below than that of the VGDD, which indicates that the VGDC-A-C could work longer when the same energy was consumed, which further means that the VGDC-A-C had higher energy utilization ratio.

#### 4.3.3 Variance of residual energy

As illustrated in Fig. 14, when the number of sensor nodes was 800 and 1600, the variance value of the VGDC-A-C was slightly larger than that of the VGDD. However, when the number of sensor nodes was 1000, 1200, and 1400, the variance value of the VGDC-A-C was slightly lower than that of VGDD. The performances of two algorithms regarding the balance of energy consumption were similar. As the network lifetime of the VGDC-A-C was larger than that of VGDD, the energy consumption balance of the VGDC-A-C was slightly better than that of the VGDD.

#### 4.3.4 Average number of transmission hops

As illustrated in Fig. 15, when the number of sensor nodes varied from 800 to 1600, the average number of transmission hops was about 4 hops in the VGDC-A-C. The corresponding number of the VGDD was greater than 7. The average number of transmission hops indicated that when the VGDC-A-C was used, and source node sent data to the mobile sink, the hops of the packet could be reduced by 3 hops compared to the VGDD, which was because the trajectory of a mobile sink was adjusted dynamically and moved toward the event area. Meanwhile, when mobile sink allocated the parking time, the sink parked longer in the grids, which processed more sensory data. By constantly adjusting the movement trajectory and parking time, mobile sink could get closer to the source nodes, and the sensory data could be uploaded to the mobile sink faster. However, the mobile sink had a predetermined trajectory in the VGDD, so the mobile sink could not adjust its moving status according to the changes in the event area. Thus, the VGDC-A-C had better real-time performance than the VGDD.

## 5 Conclusions

In this paper, the algorithm for real-time data collection for applications with centrally distributed events, called the VGDC-A-C, is proposed and analyzed. Firstly, a virtual grid virtual grid structure is introduced to initialize the network. The virtual grid structure can divide the network into several virtual square areas with the same size, where virtual grids of different areas have different RCN and DN. The structure is the basis of sensory data routing. Then, the routing of sensory data is discussed. With the help of a virtual grid structure, the sensory data can be routed to the mobile sink easily

and automatically. Afterward, the trajectory planning of a mobile sink is proposed such that the mobile sink can move closer to the event area and park in the virtual grids longer, which increases the amount of sensory data. Using the proposed algorithm, the total length of routing paths and the transmission delay are decreased. To reduce the energy consumption of updating of a mobile sink location, we propose the local updating. Finally, we proposed the re-election of head nodes in the virtual grid cells to balance energy consumption. Compared with the VGDD algorithm, the VGDC-A-C algorithm prolongs network lifetime and decreases transmission delay in the applications with centrally distributed events.

#### Abbreviations

IWSNs: Industrial wireless sensor networks; VGDC-A-C: Virtual grid-based real-time data collection algorithm for applications with centrally distributed events; MWST: Minimum Wiener index spanning tree; MST: Minimum spanning tree; RLW: Random line walk; EEGDD: Energy efficient grid-based data dissemination routing mechanism; O-LEACH: Optimizing LEACH clustering algorithm; RNS: Convergence nodes; VGDR: Virtual grid-based dynamic routes adjustment scheme; VGDD: Virtual grid-based data dissemination scheme; ID: Identification; RCN: Row column number; DN: Direction number; GS: Grid of sink; LU: Head node in the upper left corner; RU: Head node in the upper right corner; LD: Head node in the lower left corner; RD: Head node in the lower right corner; UG: Head node in the grid above; DG: Head node in the grid below; LG: Head node in the left grid; RG: Head node in the right grid; IG: Intersecting grid

#### Funding

The work is supported by "the Fundamental Research Funds for the Central Universities, no. 2017B14714", supported by "the National Natural Science Foundation of China under grant no. 61572172", and supported by "Changzhou Sciences and Technology Program, no. CE-20165023 and no. CE20160014" and "six talent peaks project in Jiangsu Province, no. XYDXXJ-S-007".

#### Availability of data and materials

The values of simulation parameters are listed in Table 6.

#### Authors' contributions

CZ, XL, GH, JJ-N, and SZ designed the study, performed the research, analyzed the data, and wrote the paper. All authors read and approved the final manuscript.

#### Authors' information

Chuan Zhu received the Ph.D. degree from the Department of Computer Science, Northeastern University, Shenyang, China, in 2009. And in December 2017, he finished his work as a Postdoctoral Researcher with Hohai University. He is currently a Lecturer in the Department of Information and Communication System, Hohai University, China. He has authored over ten papers in related international conferences and journals. His current research interests are sensor networks, cloud computing, and computer networks. Xiaohan Long is a Master degree candidate of the Department of Internet of things and its Application at Hohai University, China. His current research interests are wireless sensor networks, underwater wireless sensor networks, cloud computing, and Android security software development. Guanjie Han is currently a Professor in the Department of Information and Communication System, Hohai University, Changzhou, China. In 2004, he received the Ph.D. degree from Northeastern University, Shenyang, China. From 2004 to 2006, he was a Product Manager for the ZTE Company. In February 2008, he finished his work as a Postdoctoral Researcher in the Department of Computer Science, Chonnam National University, Gwangju, Korea. From October 2010 to 2011, he was a Visiting Research Scholar in the Osaka University, Suita, Japan. He is the author of over 230 papers published in related international conference proceedings and journals and is the holder of 100 patents. His current research interests include sensor networks, computer communications, mobile cloud computing, and multimedia communication

and security. Dr. Han has served as a Co-chair for more than 50 international conferences/workshops and as a Technical Program Committee member of more than 150 conferences. He had been awarded the ComManTel 2014, ComComAP 2014, Chinacom 2014, and Qshine 2016 Best Paper Awards. He is a member of IEEE and ACM.

Jinfang Jiang is currently a Lecturer in the Department of Information and Communication System at Hohai University, China. She received her Ph.D degree in Information and Communication Engineering from Hohai University, China, in 2015. Her current research interests are security and localization for sensor networks.

Sai Zhang received the Master degree from the Department of Information and Communication System at Hohai University, China, 2017. He has published 1 paper in related international conferences and journals. He is working at Huawei Technologies Co., Ltd. currently.

#### Competing interests

The authors declared that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 11 January 2018 Accepted: 1 May 2018

Published online: 29 May 2018

#### References

- L Shu, M Mukherjee, X Wu, Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks. *IEEE Commun. Mag.* **54**, 22–28 (2016)
- L Shu, M Mukherjee, X Xu, K Wang, X Wu, A survey on gas leakage source detection and boundary tracking with wireless sensor networks. *IEEE Access.* **4**, 1700–1715 (2016)
- B Ahmed, B Walid, R Herve, Optimal WSN deployment models for air pollution monitoring. *IEEE Trans. Wireless Commun.* **16**, 2723–2735 (2017)
- M Amjad, L Jaime, S Sandra, A secure and low-energy zone-based wireless sensor networks routing protocol for pollution monitoring. *Wireless Commun. Mobile Comput.* **16**, 2869–2883 (2016)
- V Carlos, D Yezid, Delay/disruption tolerant network-based message forwarding for a river pollution monitoring wireless sensor network application. *Sensors.* **16**, 436–460 (2016)
- CZ Zulkifli, HN Hassan, W Ismail, et al, Embedded RFID and wireless mesh sensor network materializing automated production line monitorin. *ACTA Phys. Polonica A.* **128**(B86-B89) (2015)
- G Han, L Liu, S Chan, R Yu, Y Yang, HySense: a hybrid mobile CrowdSensing framework for sensing opportunities compensation under dynamic coverage constraint. *IEEE Commun. Mag.* **55**, 93–99 (2017)
- G Han, L Zhou, H Wang, W Zhang, S Chan, A source location protection protocol based on dynamic routing in WSNs for social internet of things. *Futur. Gener. Comput. Syst.* **82**, 689–697 (2017)
- X Tian, Y Zhu, K Chi, J Liu, D Zhang, Reliable and energy-efficient data forwarding in industrial wireless sensor networks. *IEEE Syst. J.* **11**, 1424–1434 (2015)
- G Han, X Yang, L Liu, W Zhang, A joint energy replenishing and data collection algorithm in wireless rechargeable sensor networks. *IEEE Internet Thing J.* (2017)
- E Lee, S Park, J Lee, Novel service protocol for supporting remote and mobile users in wireless sensor networks with multiple static sinks. *Wireless Netw.* **17**, 861–875 (2011)
- S Jannu, P Janam, A grid based clustering and routing algorithm for solving hot spot problem in wireless sensor networks. *Wireless Netw.* **22**, 1901–1916 (2016)
- C Zhu, G Han, H Zhang, A honeycomb structure based data gathering scheme with a mobile sink for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **10**, 1–16 (2016)
- C Zhu, S Zhang, G Han, A greedy scanning data collection strategy for large-scale wireless sensor networks with a mobile sink. *Sensors.* **16**, 1432–1459 (2016)
- S Han, I Jeong, S Kang, Low latency and energy efficient routing tree for wireless sensor networks with multiple mobile sinks. *J. Netw. Comput. Appl.* **36**, 156–166 (2013)
- K Shin, S Kim, Predictive routing for mobile sinks in wireless sensor networks: a milestone-based approach. *J. Supercomput.* **62**, 1519–1536 (2012)
- G Shi, J Zheng, J Yang, Z Zhao, Double-blind data discovery using double cross for large-scale wireless sensor networks with mobile sinks. *IEEE Trans. Vehicular Technol.* **61**, 2294–2304 (2012)
- P Singh, R Kumar, V Kumar, An energy efficient grid based data dissemination routing mechanism to mobile sinks in Wireless Sensor Network, International Conference on Issues and Challenges in Intelligent Computing Techniques, 401–409 (2014)
- C Tunca, M Dönmez, S Isik, C Ersoy, Ring routing: an energy-efficient routing protocol for wireless sensor networks with a mobile sink. *IEEE Trans. Mobile Comput.* **14**, 1947–1960 (2015)
- N Wang, Y Chiang, Power-aware data dissemination protocol for grid based wireless sensor networks with mobile sinks. *IET Commun.* **5**, 2684–2691 (2011)
- S Mottaghi, M Zahabi, Optimizing LEACH clustering algorithm with mobile sink and rendezvous nodes, *AEU-Int. J. Electron. Commun.* **69**, 507–514 (2015)
- C Konstantopoulos, G Pantziou, D Gavallas, et al, A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks. *IEEE Trans. Parallel Distributed Syst.* **23**, 809–817 (2012)
- A Khan, A Abdullah, M Razzaque, VGDR: a virtual grid-based dynamic routes adjustment scheme for mobile sink-based wireless sensor networks. *IEEE Sensors J.* **15**, 526–534 (2015)
- A Kinalis, S Nikolettseas, D Patroumpa, Biased sink mobility with adaptive stop times for low latency data collection in sensor networks. *Inf. Fusion.* **15**, 56–63 (2009)
- S Ghafoor, M Rehmani, S Cho, An efficient trajectory design for mobile sink in a wireless sensor network. *Comput. Electr. Eng.* **40**, 2089–2100 (2014)
- G Yang, S Liu, X He, N Xiong, Adjustable trajectory design based on node density for mobile sink in WSNs. *Sensors.* **16**, 2091–2114 (2016)
- A Khan, A Abdullah, M Razzaque, VGDD: a virtual grid based data dissemination scheme for wireless sensor networks with mobile sink. *International J. Distributed Sensor Netw.* **11**, 1–17 (2015)
- G Han, X Yang, L Liu, W Zhang, M Guizani, A disaster management-oriented path planning for mobile anchor-based localization in wireless sensor networks. *IEEE Trans. Emerging Topics Comput.* (2017)

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)