Li *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:166
https://doi.org/10.1186/s13638-020-01767-5

EURASIP Journal on Wireless Communications and Networking

RESEARCH **Open Access**

Bi-adjusting duty cycle for green communications in wireless sensor networks

Guopeng Li[1], Fufang Li[2*], Tian Wang[3], Jinsong Gui[1] and Shaobo Zhang[4]

* Correspondence: liff@gzhu.edu.cn
[2]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China
Full list of author information is available at the end of the article

Abstract

Green communications is a challenging issue for communications and networking. In order to save energy, nodes of the wireless sensor networks (WSNs) usually adopt the low duty cycle mode which brings a large delay to the event detection and data transmission to the sink and deteriorates the network's timely processing of the event. In this paper, bi-adjusting duty cycle schedule (BADCS) scheme is proposed to reduce event detection latency as well as data routing delay for low duty cycle wireless sensor networks. BADCS scheme consists mainly of two duty cycle adjustment algorithms: (a) active slot asynchronous adjustment algorithm for nodes in the same sensing area; (b) the continuous adjustment algorithm for two adjacent nodes on the routing path with one active slot interval. Specifically, the operations are 2-fold. First, perform asynchronous operations on the active slots of the nodes in the same sensing area, so that the active slots of the nodes in the same sensing area are distributed as evenly as possible without overlapping. In this way, it is possible to reduce the latency by the time the event is perceived after its occurrence. Secondly, active slots of the nodes on the routing path are adjusted to be with pipeline style, so that when the nodes receive the data packet, they can route through the continuous active slots, thus greatly reducing the delay of data routing. Two adjustment algorithms of active slot are given in detail in this paper. The performance of BADCS mode is discussed in detail, and its performance is better than the previous strategy. Comprehensive experiments are conducted, and the results demonstrate that the proposed BADCS scheme significantly improves event detection performance in terms of detection latency, detection probability, and routing delay. Detection delay and routing delay are reduced as high as 3.91% and 56.22% respectively.

Keywords: Wireless sensor networks, Low power, Duty cycle, Event detection latency, Routing delay

1 Introduction

Green communications refers to a new generation communication conception that can save resources, reduce environmental pollution and environmental waste, and lessen harm to the human body and the environment from communication equipment. At present, innovative design of green communication network is mainly aimed at network deployment, network management, and wireless resource management. Wireless sensor networks (WSNs) are an important part of the Internet of Things (IoTs), which are composed of tiny wireless sensing nodes consisting of sensing device, data processing device, memory, battery, and communication device [1–5]. With the rapid development of microelectronic device technology, the processing power, sensing ability, and sensing accuracy of sensor nodes are getting higher and higher. Computing power of current sensor nodes had already exceeded computer a decade ago. On the other hand, the manufacturing cost of sensor nodes and the volume of the device become increasingly smaller. With the development of artificial intelligence technology and big data network, WSNs have gained new vitality and thus applied to increase fields [6–8]. One of the important applications of WSNs is the monitoring of events [9–11]. For example: sensor nodes are deployed in key places, such as treasury, to protect the important infrastructure by sensing various physical phenomena such as vibration, fire, and sound of the surrounding environment. In forest fire monitoring, wildlife protection, WSNs also play an important role. By deploying sensor nodes in the forest, the fire is detected in time to win time to minimize disasters [12–14].

Research areas of WSNs are really wide, such as energy consumption [15, 16], lifetime [16], and network security [17–20]. This paper focuses on the important impacts of the following key research issues directly related to event detection performance on WSNs [21–24]:

(1) The latency for sensor node detection the event is called event detection latency (EDL). EDL refers to the time elapsed after the event occurs and is sensed by the sensor nodes. Obviously, the smaller the EDL, the better.
(2) The delay that sensing event’s data routing to sink is called data routing delay (DRD). DRD refers to the time that a sensor node perceives an event and generates a data package embracing the event information until the data package is routed to sink. Obviously, the smaller the DRD, the better.
(3) Lifetime generally refers to the time when the first node in the WSNs dies. Obviously, the longer the lifetime, the better.

However, enabling EDL and DRD as small as possible and the lifetime as long as possible is a challenging issue [21–24]. Because whether it is to shorten the EDL or DRD time, it is necessary to increase the active (i.e., running) time of the sensing node or the transmitting node, which consumes a large amount of energy of the WSNs node, which causes the node to fail due to energy exhaustion and depletion, which in turn leads to a shortened lifetime of the WSNs. In other words, shortening the EDL or DRD time means shortening the lifetime of WSNs and vice versa. In WSNs, sensor nodes are powered by batteries. At the same time, sensor nodes are generally deployed in wild places. Therefore, it is unrealistic to charge the drained nodes [25–28]. This is because in practical application scenarios, the number of WSN nodes is huge. On the one hand, the deployment of charging facilities for a large number of nodes is very difficult because the node locations are difficult to reach. On the other hand, it takes considerable time to charge each node. The time it takes to charge a large number of nodes is unacceptable, and it is almost impossible to achieve. Therefore, how to extend lifetime becomes a key research topic of WSNs when node energy is limited. Researchers found that more than 70% of the sensor node energy is consumed by communication [3, 8, 18]. To maximize the lifetime of the wireless sensor network (WSN) which is a green computing problem of WSN, Khatri et al. [29] proposed a balance-based WSN lifecycle maximization framework, namely, the development of energy-based shift (E-shift) and load-based. The L-shift algorithm is used to maximize the lifetime of the network. Analysis and simulation experiments show that their method has significant advantages compared with existing similar methods. Considering that the existing research work has insufficient effects on the impact of multiple parameters on energy consumption, Aanchal et al. [30] studied the WSN life cycle maximization technology based on Huffman coding and ant colony algorithm. Their method considered the units of hops, path length, load residual energy, and minimum energy node of the path node. Due to the use of advanced ant (a-ant), regression ant (r-ant) algorithm, and Huffman coding to identify and select the best path, their method is superior to the existing methods in many important performance indicators, such as average residual energy, energy consumption, number of living sensors, and standard deviation of energy. Communication consumption consists of the following parts: energy consumption for receiving and sending data and energy consumption for idle waiting and energy consumption for sleep. The energy consumption when the node sends data, receives data, and waiting is several orders of magnitude more than the energy consumption of the node when it sleeps. Therefore, in order to save battery energy, the node should be kept in sleep state as much as possible. The duty cycle method is such a way to effectively save node energy [31–33]. In such a method, sensor nodes divide time into an equal period of time, which is composed of multiple slots. The node selects one slot to be in the active state, and the other slots to sleep to save energy. The duty cycle also refers to the ratio of the time the node is in the active state to the length of the entire cycle [3, 31, 33–35].

However, when the node is in the sleep state, it will deteriorate the network performance, since the node cannot be aware of the event when the node is in the sleep state, and cannot receive and send data [36–38]. If the event occurs in the sleep state of the node, the node cannot sense the event at this time, and it is necessary to wait until the sensor node switches to the awaking state to sense the event. Therefore, the longer the node sleep is, the more latency it detects the event. For the occurrence of major events, if its sensing latency is large, it may cause significant losses. Obviously, in the duty cycle-based WSNs, the larger the duty cycle of the node, the more the energy consumption of the node, and the smaller the lifetime of the node, although the latency of the node detection event is small. And if the node uses a smaller duty cycle, although it can save energy, it will make the event's latency more.

On the other hand, the node's duty cycle method also affects the data routing delay. In a network where the node is always active, when the node has data to send, it can select a relay node from so called forwarding node set (FNS) whose nodes are closer than the sender to the sink. It can make the number of hops required to reach the sink minimum when the node closest to the sink is selected, so that the data routing delay is minimum [16, 39]. But in duty cycle-based WSNs, there is another situation. Because of the periodic awake/sleep of nodes, when sender sends data, the worst case is that all nodes in FNS are in sleep state, so sender needs to wait for one of nodes in FNS to wake up before routing. But even when the sender transmits data, there exist at least one node in its FNS in the active state, but in this case, the nodes closest to the sink in the FNS are not necessarily in the active state [16, 39]. When sender chooses relay node from FNS may be far from sink, which will make the distance of one-hop routing to sink direction smaller, and more hops are needed to route to sink, as a result, the delay increases. And if the sender waits for the nearest forwarding node to wake up, the waiting time will increase delay [16, 39]. How to maximize lifetime in duty cycle-based WSNs while minimizing event detection latency and data routing delay is an issue with the challenge [16, 39].

Some studies have been carried out to reduce delay. It is a very effective method to reduce delay by increasing the duty cycle. From the previous discussion, when the duty cycle of the node increases, the event detection latency and data routing delay become smaller. However, increasing the duty cycle will increase the node's energy consumption and thus reduce the lifetime, so this method reduces the delay at the expense of life. Therefore, a lot of research in reducing delay under the condition of not reducing lifetime, obviously this is a better way. Some studies have proposed adjusting the active slots between nodes in the sensing range to make these active slots evenly distributed to reduce event detection latency [40]. However, some studies have proposed adjusting the active slots of nodes on the routing path to be continuous, so that packet data can be continuously performed during routing, which reduces the data routing delay [41].However, there is no overall study on delay optimization, so we propose a bi-adjusting duty cycle schedule (BADCS) scheme which can simultaneously reduce event detection latency as well as data routing delay for low duty cycle wireless sensor network (WSNs). The main innovations of this work are:

(1) A bi-adjusting duty cycle schedule (BADCS) scheme is proposed to reduce event detection latency as well as data routing delay. The BADCS scheme consists mainly of two duty cycle adjustment algorithms: (a) active slot asynchronous adjustment for nodes in the same sensing area and (b) the connection adjustment algorithm for two adjacent nodes on the routing path with one active slot. First, perform asynchronous operations on the active slots of the nodes in the same sensing area, so that the active slots of the nodes in the same sensing area are distributed as evenly as possible without overlapping. In this way, it is possible to reduce the latency by the time the event is perceived after its occurrence. Secondly, the active slots of the nodes on the routing path are adjusted to be sequentially arranged, so that when the nodes receive the data packet, they can route through the continuous active slots, thus greatly reducing the delay of data routing.

(2) The performance of the BADCS mode is analyzed in detail, and its performance is better than the proposed strategy. Comprehensive experiments are conducted, and results demonstrate that the proposed BADCS scheme significantly improves event detection performance in terms of detection latency, detection probability, and routing delay. It reduces as high as 5.3% of detection delay and 53.7% of routing delay.

The rest of this paper is organized as follows: in Section 2, the related work is stated. In Section 3, the network model and problem statement are introduced. Then, BADCS scheme is illustrated in Section 4. The theory analysis result of BADCS is presented in Section 5. The experimental comparison results are presented in Section 6. Finally, Section 7 provides conclusions.

2 Background and related work

Wireless sensor networks (WSNs) are often deployed in battles, fire-prone areas, rare animal protection, and monitoring of emergency event or target [20, 21]. In such application scenarios, when the event or target appears, it is necessary to quickly monitor the occurrence of the event or target and deliver the data to the sink as fast as possible; otherwise, it may incur significant damage and disaster [7, 20, 21]. Therefore, some researchers have proposed various strategies for how to reduce event detection latency and data routing delay. This section will mainly review the related research on this paper.

2.1 Research on reducing event detection

The reason for the event detection latency (EDL) is that the sensor nodes work with a duty cycle that has the following properties: when the event occurs, all nodes may be in the sleep state, and the event cannot be detected. When there is a node switch from the sleep state to the active state, the event can be monitored, and the period from the occurrence of the event to the event be monitored by the node is EDL [3, 31, 33–35]. Apparently, the size of the duty cycle will affect EDL. If the duty cycle is small, that is to say, when nodes are in the active state for a long time, the probability of event detection will be higher, so EDL will be smaller. In the case of the same duty cycle, the impact range of the event (meaning that the nodes in this range can monitor the event) is large, then the EDL is small, and the main reasons are the larger the impact range of event is, the more nodes can detect the event. And the event is not monitored when an event occurs because all nodes in this monitoring range are in sleep state, so event cannot be monitored. Therefore, the more nodes that can detect the event, the smaller the probability that these nodes are in the sleep state at the same time, so the EDL is smaller. At the same time, it also shows that the higher the density of deployed nodes, the smaller the EDL, but this will improve the cost of deploying the network [3, 31, 33–35].

According to the above analysis, the range of the event is determined by the nature of the event, and its range is difficult to adjust [42–46].. For example, to perceive whether there are enemy tanks in the battlefield environment, vibration sensors are generally used in the monitoring network through which the personnel pass. The vibration range is determined by the properties of the monitoring object and the propagation properties of the vibration wave, which cannot be controlled [47–51]. Therefore, it is possible to reduce the event detection latency by increasing the node’s duty cycle and node density. However, considerable research has been done to reduce the event detection latency by adjusting the node's duty cycle and node density [52, 53].

Hu et al. [36] cleverly proposed a method to improve the quality of monitoring without reducing the network lifetime. The method they use is to use the wireless sensor network “multi-to-one” data collection process, the energy consumption of the near-sink area node is high, and the energy consumption of the low-sink area is low. Therefore, the duty cycle of the node in the far sink area is appropriately increased, so that the energy can be fully utilized, and monitoring quality can be improved without reducing the network life.

In fact, the event detection discussed above belongs to a non-cooperative monitoring mode. In the non-cooperative monitoring mode, each node monitors event independently. This mode of monitoring is suitable for applications where the position of the event is stationary. For instance, fire monitoring belongs to such a situation. When a fire occurs, the nodes that monitor the fire are independent monitoring, and the alarm information generated is sent to sink. Some studies use different priority groups according to the strength of the event signal received by the node to ensure that the data of the node with strong signal is first sent to the sink, the sink is connected to the edge network [54–56], or sent to the cloud network for processing [57–59]. Some studies use sensing information for data fusion before sending it to sink to reduce energy consumption and improve network lifetime [60, 61], but these methods are not collaborative monitoring. The main goal of these methods is to make the information of event perception accurate, the data with large information transmit first [62, 63], to reduce the amount of data, or to improve the network life.

In the mobile target detection, the collaborative monitoring method is often adopted, that is the nodes need to cooperate to complete the monitoring of the mobile target. Since the target to be monitored in such studies is mobile, when a target is detected by a node, it is necessary to predict the region to which the target will arrive at the next slot, and then increase the duty cycle of the node in the above region to reduce EDL. For instance, make its duty cycle is 1, in this way, once target enters the expected monitoring area, it can be immediately monitored, so that EDL is 0. Research based on the above ideas can be noted in [16, 52]. The advantage of such studies is to adopt a lower duty cycle in the region nodes where the target does not appear, so as to save energy, and adopt a larger duty cycle in the region where the target appears, so as to reduce EDL. However, the key of this kind of method is to accurately predict the target movement. If the prediction is not accurate, the network performance will deteriorate. When the target is not at the predicted position, the detection quality is not improved as the duty cycles of the nodes around this position have not been changed. At the same time, the duty cycle of the nodes around the wrong predicted location increases, energy is wasted, and the network performance will deteriorate. Probability-based target prediction and sleep scheduling protocol (PPSS) is such a mobile target monitoring method [64]. In previous strategies, the failure of predicting the location of target movement will lead to the quality of target monitoring decline. Therefore, they adopted the target prediction method based on kinematics and probability. Nodes awake probability lies on the target pass probability. For regions with a high probability of target pass, the nodes being awakened with a high probability, in the opposite case, the nodes being awakened with a low probability. Because of the probability-based approach, the PPSS protocol does not predict failure in all cases and reduces the number of active nodes required to some extent. Xiao Liu et al. [65] improved the PPSS protocol. Their idea is that the data sensed in wireless sensor networks when target and event occur need to be transmitted to sink. Therefore, the energy consumption of the nodes near the sink area is large, while the energy of the nodes of the far-sink area is left. Thus, it is possible to make full use of the energy of the far-sink region node to wake up more nodes in advance in the area where the movable target may occur, thereby improving the quality of the target monitoring without reducing the network lifetime.

Hu et al. [8] studied the relationship between monitoring frequency and monitoring quality in monitoring the mobile target. In previous duty cycle WSNs, a time period of a node is divided into two continuous periods: the sleep period and awake period. Thus, when the event occurs just after the node is turned into sleep, the event detection latency will be the length of the time period of sleep. In this case, because the sleep and wake time of the node is long, it will lead to a large EDL. However, if we divide the longer sleep and active periods into smaller periods, and alternate the sleep and awake periods, in this way, even if the event happens when it just turns into sleep, its EDL time is small, which can reduce EDL.

In a sensor network where the node is always active, if the node can cover the entire monitored area, it can be immediately monitored by the node when the event occurs, and its EDL is 0. But in the duty cycle-based WSNs is another situation, due to the periodic sleep/awake of the node, even if the sensing range of the node can cover the entire network, when the node is in the sleep state, some areas cannot be detected when the node sleeps, thus affecting the monitoring quality. Therefore, some researchers have proposed how to ensure that the time interval of the monitoring area of the cover is not monitored by the node is less than certain ratios when the node is in sleep/awake rotation. The easiest way is to increase the density of the nodes, because when the density of the node is ρ, the entire monitoring area can be covered, then for the sensor network with a duty cycle of $1/k$, the node density increases to the original k times, it is theoretically possible for each slot of the network to be covered by the node, but this way will increase deployment costs. Therefore, in order to save the network deployment cost, a certain detection of latency is often allowed, so that the number of nodes deployed can be reduced to reduce the cost. Hu et al. [66, 67] proposed a novel heuristic subtraction deployment strategy that does not increase deployment costs but also improves monitoring quality. In their proposed strategy, because the energy consumption of nodes in far sink area is low and surplus, the method of reducing the number of deployed nodes and increasing duty cycle can be proposed to ensure that the deployment cost is reduced without decreasing the quality of monitoring.

In the above research methods, the proposed scheme mainly by increasing the duty cycle to improve monitoring quality. The following research does not change the duty cycle of the node, but only changes the active slot of the node to reduce the EDL. The basic idea of this method is that when an event occurs, there may be multiple nodes that can monitor the event. If the active slots of these nodes are synchronized, the event will not be perceived until the active slot arrives after the event occurs. If the active slots of the nodes in the event perceptions range are staggered, the delay between event occurrence and monitoring by one of the nodes will be smaller that will help to reduce EDL. For this kind of research, see Ref. [38].

2.2 Research on data routing delay optimization

Data routing delay (DRD) is another kind of delay in event monitoring. The EDL discussed above only represents the delay from the occurrence of an event to the time it is monitored, which mainly depends on the density of nodes and duty cycle. DRD refers to the time that the packet routing that detects the event occurs from the sensor node to the sink node. The main factors affecting DRD are as follows: duty cycle, node density, order of active slots, routing strategy, etc. The above factors are discussed separately below.

The impact of duty cycle on DRD is as follows: the node that monitors the event is called source node, and the generated data needs to go through multi-hop route to reach the sink, so that it is known by the control center, and then take measures to deal with the event in time. In fact, since the data packet containing the event occurrence message reaches the sink from the source node through multiple k hops, the delay of each hop is at the same level as the event detection latency. Therefore, the data routing delay has a greater impact on the quality of event monitoring. The main reason for data routing delay is that when a data packet (sender) is sent to the next hop node, the receiver node may be sleep because the node uses the duty cycle mode, so the sender needs to wait for the receiver to be awake before routing, which will result in delay. The worst case is when the sender has data to send, the receiver is about to sleep, in which case the delay is the largest. In order to decrease DRD, many researchers have also proposed some research. We make the following review:

(1) Reduce DRD by selecting the appropriate node. In the routing process, when the sender forwards data, there are often multiple next hop nodes to choose from. These nodes that are closer to sink than sender and are within the sender's transmission radius are called sender's forwarding nodes set (FNS). Obviously, when the sender has data to transmit, the nodes closest to the sink should be selected, because the data packet is forwarded farther away, so that the sink can be reached with the fewest hops. However, when the sender has data to transmit, the node closest to the sink is not necessarily in the active state, so the sender cannot get the optimal relay node. At this time, the sender has two choices: the first choice is to select the forwarding node that is in the active state to transmit as the relay node. In this case, although the distance traveled to the sink is not the largest each time, since there is no waiting time, this scheme may route more hops, but the delay per hop is small. The second choice is that the sender continues to wait for the node closest to the sink to wake up and then selects this node as the relay node. The advantage of this scheme is that the number of hops required for routing is small, but the delay per hop may be larger. Based on the above analysis, Ref. [16] proposed a multi-objective optimization of the relay node selection strategy to improve performance.

(2) Adjusting the duty cycle of the node. Similarly, increasing the duty cycle of the node can directly and effectively reduce the DRD. Similarly, increasing the duty cycle of the node can directly reduce DRD. Because the larger the duty cycle of the node is, the smaller the probability of relay node being in sleep state, thus reduce delay. However, the disadvantage of increasing the node duty cycle is that it increases the energy consumption of the node, which reduces its lifetime. Hu et al. [36] proposed an adaptive duty cycle strategy to reduce delays. In their strategy, the end-to-end delays are reduced by increasing the duty cycle of nodes that energy surplus and away from the sink. And

the advantage of this strategy is that it reduces the delay without reducing the network lifetime.

(3) Reduce DRD by adjusting the order of node active slots. The main idea of this method is as follows. The main reason for the large delay in the routing process is after the sender receives the data in its active slot, it will send it to the next hop in its next slot (set to the i -th slot). If the active slot of its next hop node happens to be the i -th slot, then the delay in this case is the least. Conversely, if the distance between the active slot of the next hop node and the i -th slot is k slots (i.e. $i + k$-th slot), then the next hop node needs to wait for k slots to receive data; in this case, the delay is larger in this case. However, if the active slot of the node can be adjusted so that the active slot of the adjacent nodes in the routing path can be arranged continuously, the DRD will be smaller. Such a study can be seen at Ref. [39].

Although numerous studies have been proposed to reduce EDL and DRD, to the best of our knowledge, there is no research that can effectively combine these two kinds of delays. For example, by adjusting the active slot strategy is a relatively good strategy without affecting lifetime, but the present research is independent, and the two strategies are not combined organically, so that the system performance is not very good. Therefore, this paper proposes a comprehensive and effective method to reduce delay.

3 Network model and problem statement

3.1 The network model

The network model used in this paper is as follows. There is a sink node in the network, with the sink node as the center of the circle, and n identical sensor nodes are randomly located in a two-dimensional fan-shaped planar network. The network can be expressed as$\{s, v_1, v_2, v_3, v_4 \ldots v_n\}$. v_i is the id of the sensor node, the transmission radius of the sensor node is R_t, and the sensing radius is R_s.

The state of each node can be active or sleep, and each sensor node wakes up only once in a cycle. The function of the sensor node is to sense events, exchange information with neighbor nodes, and calculate data. The sensor node is energy limited, and the sink nodes are always awake, with unlimited energy, can receive, send information, and calculate data at any time. There are m slots in each cycle, and the sequence number of the time slot is $0 \sim m$-1. At the initial network, the active slot of the sensor node is randomly set. Assume that after the fire occurs, the node can monitor the event in the same slot. When the event occurs in 3 slots, if the node wakes up in 3 slots, the event detection delay is 0. If the node wakes up in 4 slots, the event detection delay is 1. In the transmission process, 1 hop needs 1 slot, and the successor node can receive the data at the earliest slot in the same period in which the predecessor node starts transmitting data or the first slot in the next cycle. For example, node 1 sends a packet at 3 slots. If node 2 is in an area with node 1 as the center and transmission range as the radius, the distance between two nodes is 1 hop. If node 2 wakes up at 4 slots, data routing delay is 1 slot; if node 2 wakes up at 5 slots, data routing delay is 2 slots. The initial wireless sensor network model is shown in Fig. 1, and the number in the node represents the initial active slot.

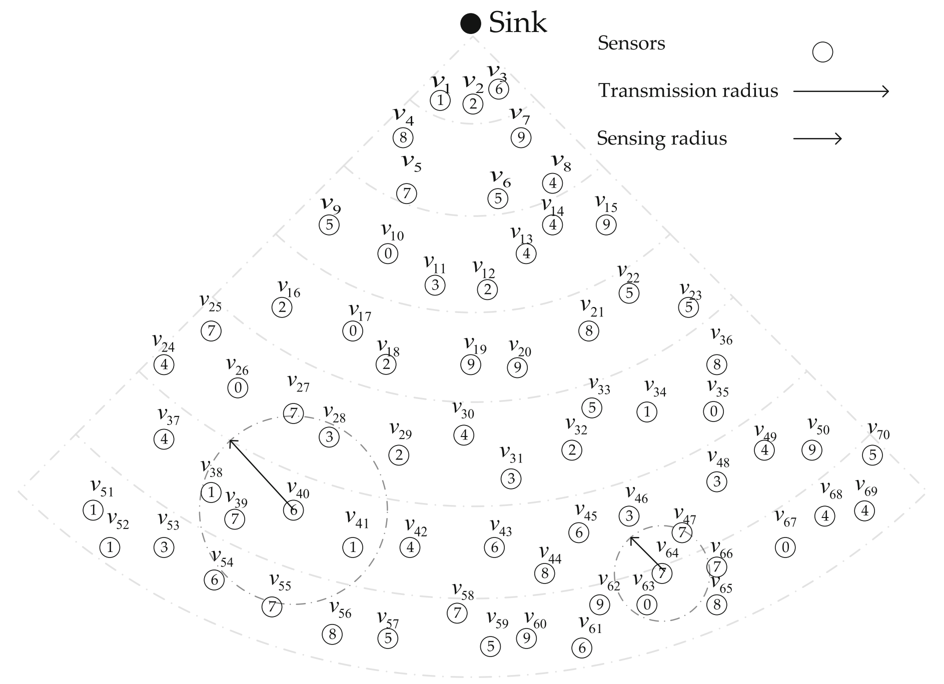

Fig. 1 Network structure diagram

3.2 The problem statement

3.2.1 Definition 1

Minimizing total delay in wireless sensor networks without reducing lifetime. The total delay D consists of two parts: the first part is the event detection latency (EDL), which refers to the time elapsed after the event occurs and is sensed by the sensor nodes. The second part is data routing delay (DRD), which refers to the time when a sensor node perceives an event and generates a data package embracing the event information until the data package is routed to sink. Using S to represent EDL and using T to represent DRD, the formula for calculating total delay is as follows:

$$\textit{total delay}\ D = S + T \tag{1}$$

The aim of this paper is to reduce and minimize the total delay of WSNs network; it can be calculated as follows.

$$\min(\textit{total delay}) = \min\left(\sum_{i\subseteq route} t_i + \sum_{j\subseteq event} s_j\right) \tag{2}$$

In formula (2), the expression term of min(*total delay*) represents to minimize the total delay. The expression term of $\sum_{i\subseteq route} t_i$ represents the summation of the data routing delays on the path for each path in the WSNs. Since the duty cycle is not adjusted, it will not affect the lifetime of the WSNs network.

As is mentioned in Sections 1 and 2, shortening the EDL and DRD time and extending the network survival time are a pair of contradictory issues in the research field of WSN networks. Although many researchers have proposed many methods to reduce EDL and DRD, these methods either sacrifice the lifetime of the WSN network or the proposed method only focuses on one of the reduction indicators of EDL or DRD which still cannot meet the needs of practical applications. Therefore, it is an urgent and important scientific problem in the area of WSN network research to explore innovative methods to effectively shorten EDL and DRD without reducing the lifetime of WSN network, which is the main research work of this article.

The main notations which would be adopted later in this paper are listed in Table 1.

4 The design OF BADCS scheme

4.1 Research motivation

Specifically, some studies have studied ways to reduce event detection latency without reducing lifetime [8], and the method they use is to reduce the delay by properly arranging the node's active slot. When an event occurs, for a dense WSN network, there are multiple nodes in the scope of the event, among which only one node is needed sensing events. In this case, obviously, if the active slot of the node within the scope of the event is in the same slot, the expectation of its event detection latency is the largest. If the active slots of each node are different, the special case is that the number of nodes in the event perception range is n, and the number of slots in a time period of the node is n; in this arrangement, if each node's active slot is different, the event detection latency is 0 regardless of which slot event occurs. Therefore, if the active slot of the nodes in the event monitoring range is evenly distributed, the event detection latency can be minimized; based on this principle, some researchers have proposed corresponding algorithms.

In the WSNs of this paper, after the fire event occurs, the sensor within the range of the sensing radius of the wireless sensor can sense the fire event and generate data. The data package is then transmitted to the sink node through a pre-arranged network, so that the purpose of fire monitoring can be achieved.

Since the wireless sensor network is limited in energy, different sensors have different active slots in one cycle, so there will be delays in the sensing process and the transmission process. In order to reduce EDL, the active slots of each node and the nodes within its sensing range are asynchronous; that is, each node wakes up in different slots, rather than randomly. In order to reduce DRD, the active slots of each node and its successor nodes within the transmission radius are as continuous as possible, rather

Table 1 Main notations adopted in this paper

Notation	Description
R_s	The sensing radius
R_t	The transmission radius
S	The event detection latency (EDL).
T	The data routing delay (DRD).
m	The cycle length
φ_c	The active slot of the current node
φ_t	The active slot of the target node
M	The maximum number of slots
d	The distance between the other nodes and the current node
ξ	The number of neighbors of the current node (including itself)

than randomly set. Through the following two examples of the adjustment process, it can be stated that this adjustment method can effectively reduce the delay.

Figure 2 (Note: please ignore the brackets and the numbers in brackets) shows WSNs with high node density, and the active slots of nodes are randomly allocated. For different fire spots and different fire slots, we calculate the expectation of event detection latency in the case of randomly allocated active slots. The time period from the fire point p_i to the time when the sensor senses the event is called EDL S, and the calculation method of the event detection latency $E_S^{p_i}$ is as follows:

$$E_S^{p_i} = \frac{\sum_{c=0}^{M} S_c}{m} \tag{3}$$

The symbol term S_c denotes the event detection latency of fire when c slot occurs. Fire may occur in any slot in the cycle. Therefore, the minimum value of c is 0 slot, i.e., fire occurs in 0 slot, the maximum value is M, i.e., fire occurs in M slot, and then the expectation of detection delay is obtained.

Calculated by the formula, $E_S^{p_1} = 0.3$, $E_S^{p_2} = 1$.

Through the adjustment of the active slot of the node, there is no node that wakes up at the same slot within the sensing range of the node, and the adjusted network situation is as shown in Fig. Error! Reference source not found. (note: for the node with bracket in the figure, the number of the node is the number in brackets).

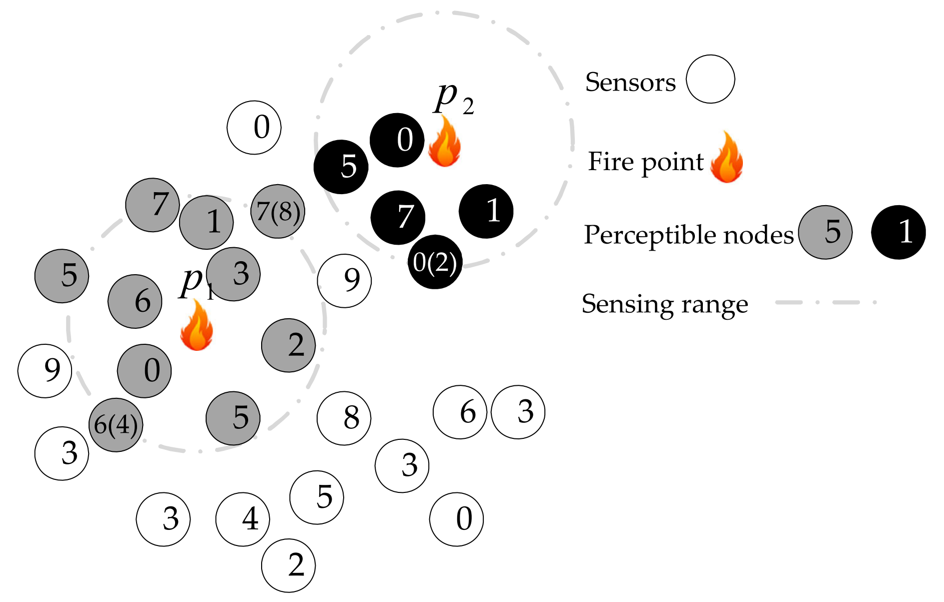

Fig. 2. The WSNs with high node density, the active slots of nodes, are randomly allocated (Note: please ignore the brackets and the numbers in brackets), and the sensor network with no nodes waking up in the same slot within the node's sensing range (Note: for the node with bracket in the figure, the number of the node is the number in brackets)

It is calculated that the expected S for monitoring the fire is $E_S^{p_1{}'} = 0.1$ and $E_S^{p_2{}'} = 0.7$. By comparing the expected values in the two cases, we can conclude that adjusting the active slot of the nodes in the sensing range can reduce EDL.

The following example is to illustrate that adjusting the active slot of adjacent nodes on the same route to continuously can reduce the data routing delay T. In Fig. 3 *(Note: please ignore the brackets and the numbers in brackets)*, there are three fire points in the wireless sensor networks, p_1, p_2, p_3. Assume that during the establishment of the routing process, the predecessor node selects the node farthest from itself as the successor node within its transmission range, forming the route shown in Fig. 3 *(Note: for the node with bracket in the figure, the number of the node is the number in brackets)*. Three fire points p_1, p_2, p_3 fire in 9 slots, 5 slots, and 3 slots. Calculate the T of data packet to sink node when active slot is randomly arranged, and active slot is continuous.

In the case of randomly setting the active slot, $T_{p_1} = 23$, $T_{p_2} = 9$, $T_{p_3} = 20$. After adjusting the active slot, $T_{p_1} = 4$, $T_{p_2} = 4$, $T_{p_3} = 4$.

By comparing DRD between the two cases, it can be concluded that arranging the active slots of adjacent nodes on the same route as continuously is helpful to reduce T.

In the next part, we will design the adjustment algorithms and demonstrate the algorithms.

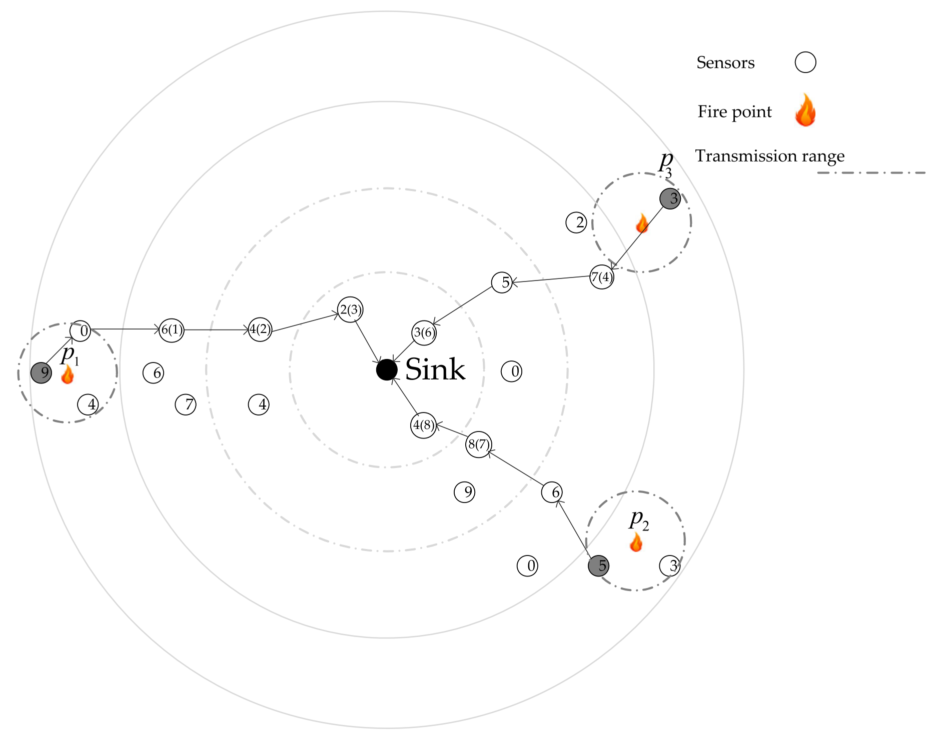

Fig. 3 The wireless sensor network with node active slot is randomly arranged (note: please ignore the brackets and the numbers in brackets), and the wireless sensor network with node active slot is continuous (note: for the node with bracket in the figure, the number of the node is the number in brackets)

4.2 Algorithm for adjusting the active slot

The previous parts introduced the research motivation of this paper. Two examples show that the operations of asynchronous and continuous adjustments for the nodes' active slots are beneficial to the reduction of EDL and DRD.

In this part, we will design the active slot adjustment algorithm and routing algorithm according to the network conditions and demonstrate the algorithms through a WSN with 70 sensor nodes.

In order to compare the delay on the initial networks with the delay of the BADCS-adjusted networks, fire points are placed at the same location of the two types of networks to simulate the fire occurrence and data transmission process. According to the simulation results, it can be concluded that after Bi-adjusting, the event detection latency can be reduced by 35.80%, for data routing delay is 45.13%, and for total delay is 43.62%; BADCS scheme is effective.

4.2.1 Asynchronous adjustment algorithm

In order to ensure that the active slots of each node are asynchronous within the sensing range, an asynchronous adjustment algorithm is designed.

The process of asynchronously adjusting the node active slot is the node obtains a random active slot during initialization. During the first cycle, the nodes interact with the neighbors to obtain and store neighbors' information. After that, the nodes that those do not satisfy the asynchronous condition send asynchronous time slot adjustment request messages to the neighboring nodes. The nodes that those receive the request messages make an adjustment action according to the data in the messages and simultaneously inform the neighbor nodes of the data of adjusted node. Until all nodes satisfy asynchronous conditions, the asynchronous adjustment ends. In order to store the information needed in the adjustment process, two tables are stored in each node. As shown in Table 2, table-a stores information such as *id*,*flag*, φ_c, ξ, *r*, and $\overrightarrow{v}$ of the current node itself.

$$flag = \begin{cases} 1, Traverse\ the\ active\ slots\ in\ table 2, no\ repeat \\ 0, Traverse\ the\ active\ slots\ in\ table 2, repeat \end{cases} \tag{4}$$

After the traversal, the repeated slots are added to the set *r*, and the value of $\overrightarrow{v}$ is assigned. The corresponding slot of the missing slot is 0, and the others are 1.

$\xi \geq m$:

Table 2 Table-a stored by sensor nodes

Notation	Description
id	The id of the current node.
flag	Binary-state variable, the flag that the node active slot is asynchronous in the sensing range.
φ_c	The active slot of the current node.
ξ	The number of neighbors of the current node (including itself).
r	A set containing repeated active slots in the sensing range.
$\overrightarrow{v}$	$\overrightarrow{v} = (v_0, v_2, v_3 \cdots v_M)$, row vector,$v_i$. The value of v_i is binary state, and v_i=1 means that a node whose active slot is i exists in the current sensing range.

The flag in table-a is a binary-state variable, and the criterion for flag 0 or 1 is as follows:
$\xi < m$:

$$flag = \begin{cases} 1, m\ sequence\ numbers\ all\ exist. \\ 0, the\ number\ of\ serial\ numbers\ is\ less\ than\ m \end{cases} \quad (5)$$

After the traversal, the repeated slots are added to the set r, and the value of $\overrightarrow{v}$ is assigned. The corresponding slot of the missing slot is 0, and the others are 1.

As shown in Table 3, table-b stores the information about the id, d, and $\phi_n$of the neighbor node of the current node that is stored (the neighbor includes itself, that is, the neighbor with d = 0). Using the table-a and table-b stored in each node, the asynchronous adjustment process is completed through the interaction of information between the nodes. The specific algorithm is described as follows.

In Algorithm 1, we have the following conventions:

1. Asynchronous slot adjustment request message (ASAM) includes the adjustment request, the node's own id, active slot, r, and $\overrightarrow{v}$
2. After each adjustment of the node's active slot, the following actions are taken:

a) Sending a slot update message to neighbors those satisfy the condition of $d \le R_s$
b) Update table-a and table-b

Algorithm 1: Asynchronous adjustment algorithm

```
1: Initialize the active slot of $n$ nodes, the period length is $m$, and the slot number is 0~$M$.
2: For each node Do
3:In the first cycle, each node exchanges data with the neighbors of $d \le R_s$, and initializes its own table-a and table-b.
4: Loop:
5:For Each slot in the next cycleDo
6:    If node $i$ 's flag = 0 then
7:      Broadcast ASAM to neighbors whose $d \le R_s$
8:    End if
9:    If Node receives ASAM && $\varphi_c \in r$ then
10:     $e$ = RandomFrom_L ($\vec{v}$) //Find the acitve slot of 0 value from $\vec{v}$ and assign slot to $e$.
11:       $\varphi_c = e$
12:       $\vec{v}$ update
13:   End if
14:node i 's flag = 1
15:End for
16:End Loop
17:If There are nodes with flag = 0 in n nodes
18: Goto Loop
19:End if
```

The asynchronous adjustment algorithm is described in the previous section, and this

Table 3 Table-b stored by sensor nodes

Notation	Description
id	Neighbor node id
d	The distance between the other nodes and the current node
φ_n	The active slot of the neighbor node

algorithm will be demonstrated in this section on the initial wireless sensor network as shown in Fig. 4 (*note: please ignore the gray color of the sensor notes).*

In the initial network, 70 sensor nodes are deployed. The numbers in the nodes represent the active slots of the nodes. They are randomly allocated. There are ten time slots in one cycle, that is, the cycle length is 10, and the time slot number is 0~9. The number of nodes waking up in each of the 10 time slots is as shown in Table 4. The initial active slot for each node in the initial network is shown in Table 5.

In the first cycle, all nodes interact with the neighbors of $d \leq R_s$ to initialize their own table-a and table-b. Since the number of nodes is large, this part of the process is not displayed.

After the two tables of the node are initialized, the nodes with flag = 0 are marked with grey colors in Fig. 4 (*note: please pay attention to the node painted with gray)*, and their table-a is shown in Table 6.

Table 6 shows the active slot, the number of neighbors, and the slot repeat set r for each node with flag 0. In the second cycle, when the node with flag 0 waking up, the asynchronous slot adjustment request messages (ASAM) are issued. That is, when the slot is 1, 4, 7, and 9, the corresponding node in the upper table sends a request message, and in other slots, no node sends ASAM. Since the data is exchanged within the sensing range, the node can receive the request sent by the neighboring node in the same time slot. Therefore, the ASAM transmission, reception, and slot adjustment can be completed simultaneously in this slot. During this cycle, the transmission and reception of ASAM and the time slot adjustment process of nodes are shown in Figs. 5, 6, 7, 8, and 9.

The adjustment process when slot = 1 is shown in Fig. 5. At this slot, v_{51} and v_{52} wake up and both flags are zero, and they send ASAM to the neighbor, while v_{53} is in the sleep state and cannot receive messages. v_{51} and v_{52} received the request from the other nodes and judged if $\phi_c \epsilon\ r$ found that the two nodes are in compliance with the adjustment conditions. They adjust their own active slots to active slots with a value of 0 in $\overrightarrow{v}$, i.e., 3 slots and 4 slots. Finally, they broadcast a message to the neighbor that they have changed the active slot and set the flag in their table-a to 1 with complete adjustment.

When slot = 4, there are two regions on the wireless sensor networks to be adjusted, which are v_{68} and v_{69} as shown in Fig. 6 and v_8, v_{13}, v_{14} as shown in Fig. 7. Slot = 4, the adjustment process of the first area is shown in Fig. 6. At this slot, v_{68} and v_{69} wake up and both flags are zero, they send ASAM to the neighbor, while v_{69} and v_{70} are in the sleep state and cannot receive messages. v_{68} and v_{69} received the request from the other nodes and judged if $\phi_c \epsilon\ r$ found that the two nodes are in compliance with the adjustment conditions. They adjust their own active slots to active slots with a value of 0 in $\overrightarrow{v}$, i.e., 3 slots and 2 slots. Finally, they broadcast a message to the neighbor that they have changed the active slot and set the flag in their table-a to 1 with complete adjustment.

Slot = 4, the adjustment process of the second area is shown in Fig. 7. At this slot, v_8, v_{13}, and v_{14} wake up and flags are zero, and they send ASAM to the neighbor, while v_6 ,v_{12}, and v_{15} are in the sleep state and cannot receive messages.

In Fig. 7, v_8, v_{13}, and v_{14} received the request from the other nodes and judged $\phi_c \epsilon$ found that the two nodes meet the adjustment conditions. They adjust their own active slots to active slots with a value of 0 in $\overrightarrow{v}$, i.e., 6 slots, 5 slots, and 0 slot. Finally, they

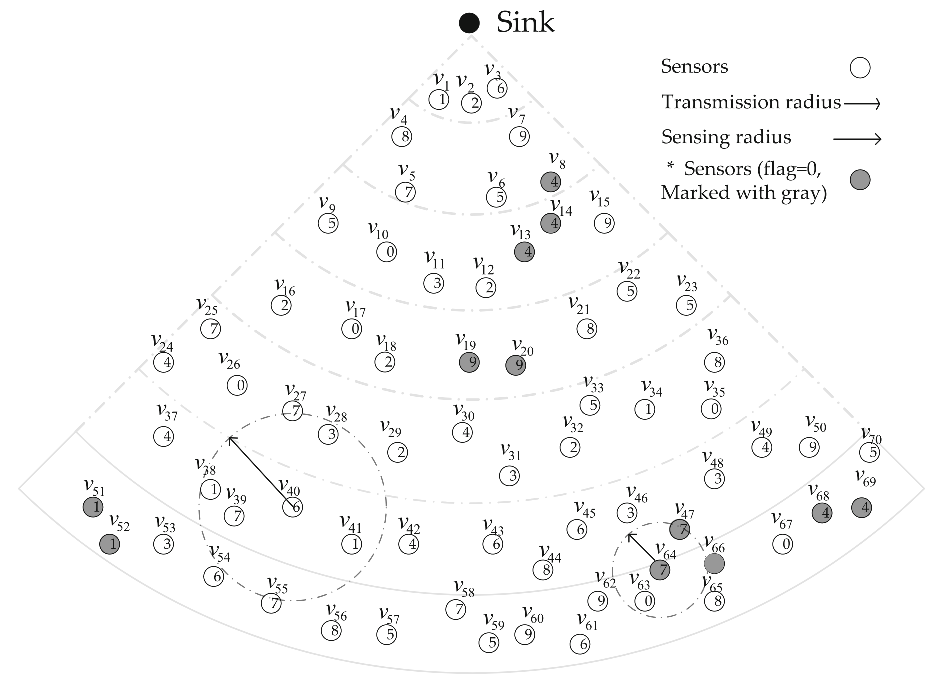

Fig. 4 Initial wireless sensor networks (note: please ignore the gray color of the sensor notes) and initial network marked with gray colors for nodes with flag = 0 (Note: please pay attention to the node painted with gray)

broadcast a message to the neighbor that they have changed the active slot and set the flag in their table-a to 1 with complete adjustment.

The adjustment process when slot = 7 is shown in Fig. 8. At this slot, v_{47}, v_{64}, and v_{66} wake up and flags are zero, and they send ASAM to the neighbor, while v_{46} ,v_{63}, and v_{65} are in the sleep state and cannot receive messages. v_{47}, v_{64}, and v_{66} received the request from the other nodes and judged if $\phi_c \epsilon\ r$ found that the two nodes are in compliance with the adjustment conditions. They adjust their own active slots to active slots with a value of 0 in $\overrightarrow{v}$, i.e., 6 slots ,5 slots, and 4 slots. Finally, they broadcast a message to the neighbor that they have changed the active slot and set the flag in their table-a to 1 with complete adjustment.

The adjustment process when slot = 9 is shown in Fig. 9. At this slot, v_{19} and v_{20} wake up and both flags are zero, and they send ASAM to the neighbor. v_{19} and v_{20} received the request from the other nodes and judged if $\phi_c \epsilon\ r$ found that the two nodes meet the adjustment conditions. They adjust their own active slots to active slots with a value of 0 in $\overrightarrow{v}$, i.e., 6 slots and 2 slots. Finally, they broadcast a message to the neighbor that they have changed the active slot and set the flag in their table-a to 1 with complete adjustment.

Table 4 The number of nodes waking up in the initial network

Slot	0	1	2	3	4	5	6	7	8	9
Number	6	7	6	6	9	8	6	9	6	7

Table 5 The initial active slot of the node

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
1	2	6	8	7	5	9	4	5	0
v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}
3	2	4	4	9	2	0	2	9	9
v_{21}	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}	v_{29}	v_{30}
8	5	5	4	7	0	7	3	2	1
v_{31}	v_{32}	v_{33}	v_{34}	v_{35}	v_{36}	v_{37}	v_{38}	v_{39}	v_{40}
3	2	5	1	0	8	4	1	7	6
v_{41}	v_{42}	v_{43}	v_{44}	v_{45}	v_{46}	v_{47}	v_{48}	v_{49}	v_{50}
1	4	6	8	6	3	7	3	4	9
v_{51}	v_{52}	v_{53}	v_{54}	v_{55}	v_{56}	v_{57}	v_{58}	v_{59}	v_{60}
1	1	3	6	7	8	5	7	5	9
v_{61}	v_{62}	v_{63}	v_{64}	v_{65}	v_{66}	v_{67}	v_{68}	v_{69}	v_{70}
6	9	0	7	8	7	0	4	4	5

So far, the second cycle has ended. Since there are no two active slot duplicate nodes or more than 10 nodes in the sensing range of a node, the flag of all nodes has been 1, and the asynchronous slot adjustment has been completed. The network after the asynchronous slot adjustment is shown in Fig. 10.

4.2.2 Routing algorithm and continuous adjustment algorithm

In the previous section, the asynchronous adjustment algorithm was described and demonstrated on the network of 70 nodes, so that the active slots of all nodes are asynchronous. In this section, we will design the routing algorithm and the continuous adjustment algorithm and demonstrate them on the network shown in Fig. 10.

The adjustment of the slot in the transmission range mainly involves the relationship between the active slots of two consecutive nodes on the same route. In order to enable the node to propagate data to the sink node with minimal cost, and to ensure that the number of successor nodes of each node is uniform, the routing of the wireless sensor network is established by algorithm 2. The algorithm 2 for establishing a route is based on the least cost routing algorithm and geographical-based opportunistic routing protocol [67]. The main process is as follows:

First, sink sets its own cost to 0, and the rest of the nodes set their own cost to the sink to ∞. The sink node first sends a route request message; after the node in the transmission range receives the route request message, a route with the sink is

Table 6 The details of the node with flag = 0

	v_8	v_{13}	v_{14}	v_{19}	v_{20}	v_{47}	v_{51}	v_{52}	v_{64}	v_{66}	v_{68}	v_{69}
id	8	13	14	19	20	4447	51	52	64	66	68	69
φ_c	4	4	4	9	9	7	1	1	7	7	4	4
ξ	3	3	4	2	2	4	2	3	4	4	3	3
r	{4}	{4}	{4}	{9}	{9}	{7}	{1}	{1}	{7}	{7}	{4}	{4}

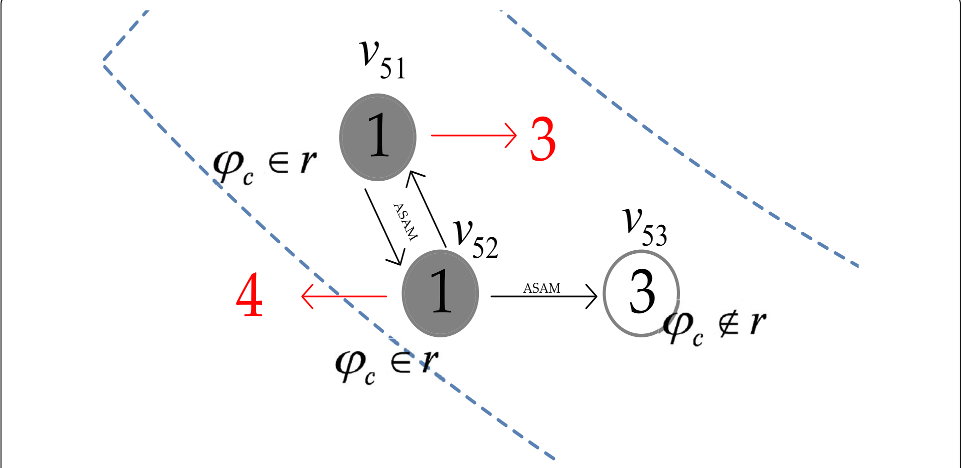

Fig. 5 Slot = 1, asynchronous time slot adjustment for v_{51}, and v_{52}

established and updates the hop count k.Thereafter, the node whose hop number is not ∞ sends the route request message, and the layer by layer progresses until the hop number of all nodes is not ∞.

When establishing a route, in order to ensure a uniform route, the node should select an appropriate node according to the established routing condition of the target node, establish a route, and send a slot update message to the transmission range.

In order to complete the establishment of the route and the continuous adjustment of the slot, information such as the hop (cost) is added to the table-a stored in each node, and the adjusted table-a is as shown in Table 7.

In the routing algorithm (algorithm 2), we have the following convention:

1. At the beginning of algorithm 2, the active slot of each node in the network is adjusted by algorithm 1

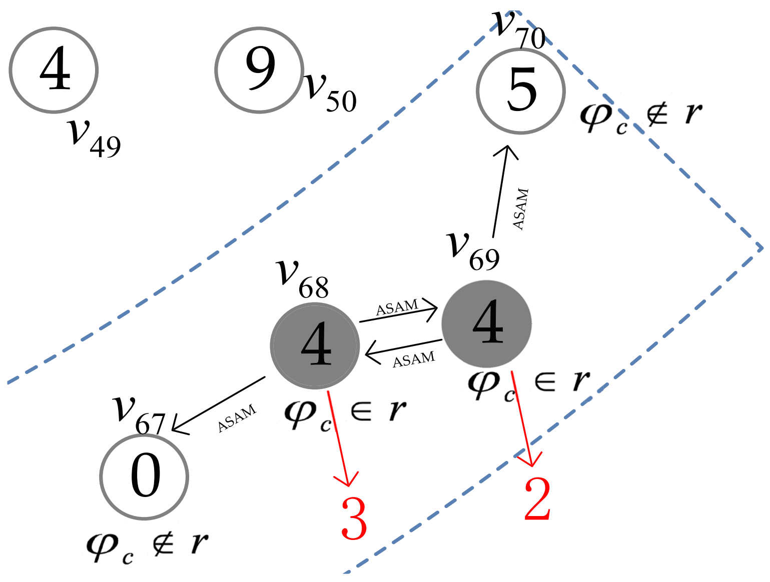

Fig. 6 Slot = 4, asynchronous time slot adjustment for v_{68}, and v_{69}

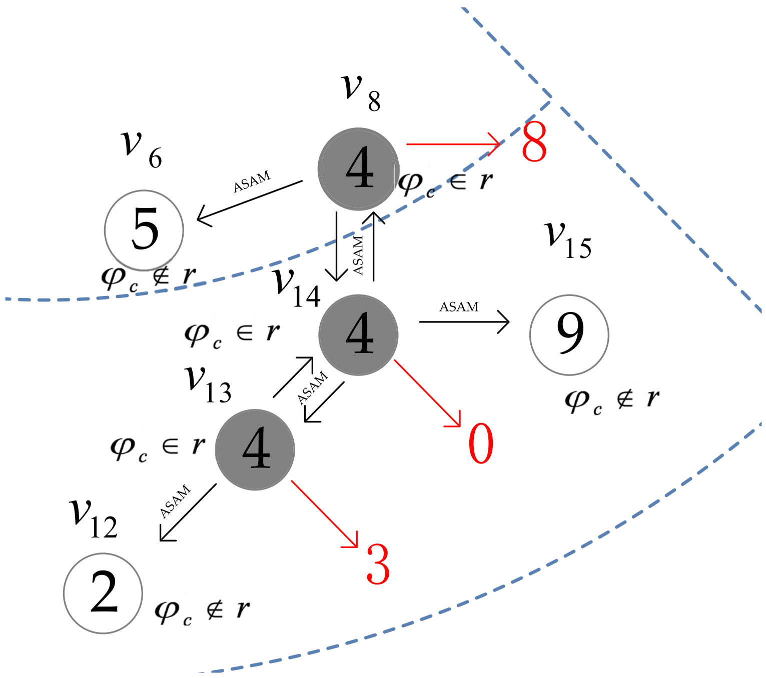

Fig. 7 Slot = 4, asynchronous time slot adjustment for v_8, v_{13}, v_{14}

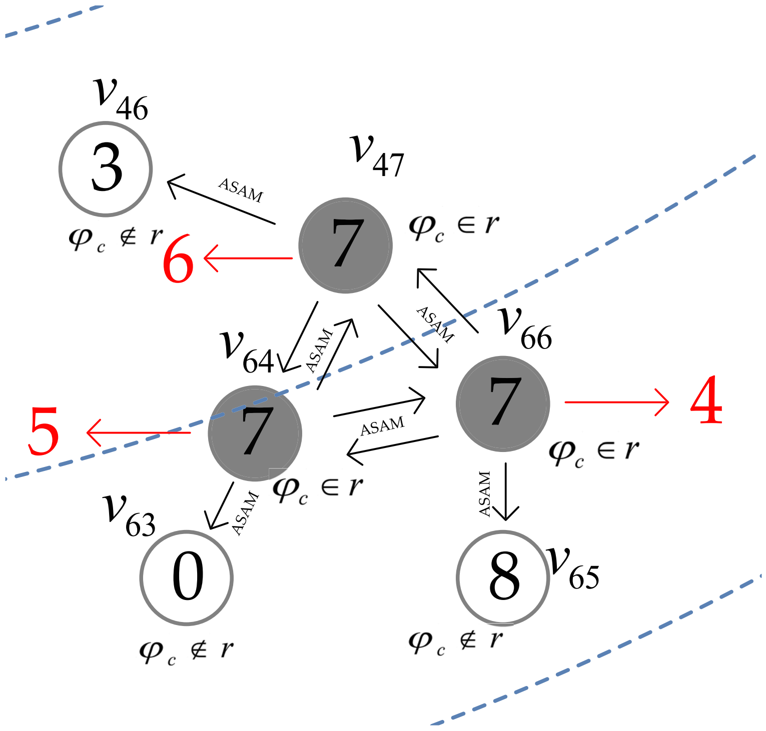

Fig. 8 Slot = 7, asynchronous time slot adjustment for asynchronous time slot adjustment for v_8, v_{13}, v_{14}

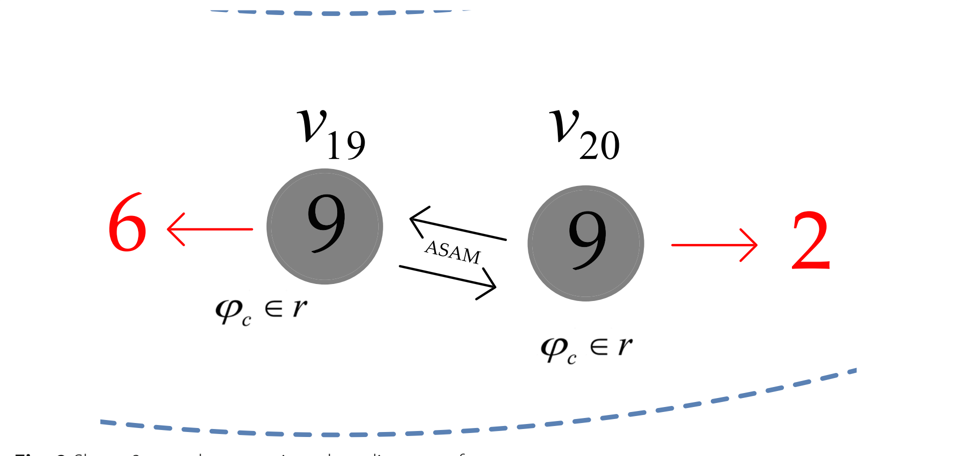

Fig. 9 Slot = 9, asynchronous time slot adjustment for v_{19}, v_{20}

2. Establishing a route request message (RRM) includes its own id, k, for convenience of description. In the algorithm, k_f, $C_n{}'$ denotes represent the node receiving k, $C_n$in RRM.
3. Creating a route message (CRM) includes two node ids of newly established routes and C_n=1 of node with smaller hop.
4. Deleting a routed message (DRM) includes two node ids that need to be deleted and C_n=− 1 of node with smaller hop.
5. k_c indicates the current node hop.

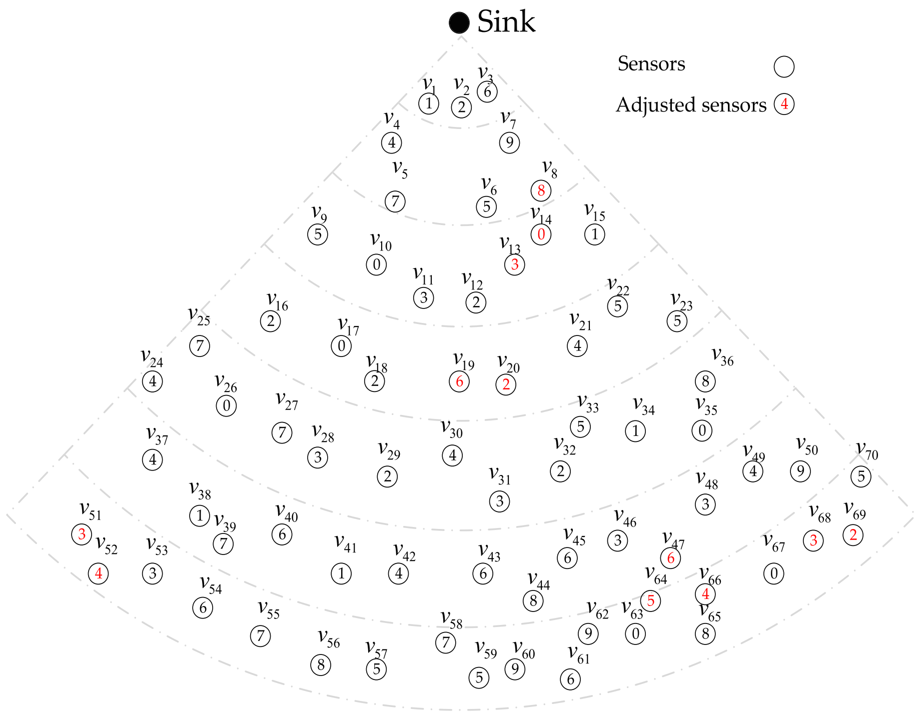

Fig. 10 The WSNs after asynchronous slot adjustment

6. $\sum C_n{}'$ indicates that the current node sums the C_n in all received packets with the same id.
7. After each adjustment of the node's active slot, the following actions are taken:

a) Sending a slot update message to neighbors those satisfy the condition of $d \le R_t$
b) Update table-a and table-b

Algorithm 2: Routing algorithm

```
1:In the current cycle, $n$ nodes exchange data with neighbors of $d \le R_t$, and initialize their own table-a and table-b.
2:In the last slot of the current cycle, the sink node broadcasts RRM
3:For Each slot of the next cycle Do
4:  If the node receives the RRM sent by the sink then
5:    $k_c = 1$
6:    $C_t = C_n{}'$
7:    Broadcast CRM
8:    $k'$=1
9:  End if
10:End for
11:Loop:
12:For each slot of the next cycle Do
13:    If the node satisfying $k$=$k'$ then
14:      Broadcast RRM
15:    End if
16:End for
17:For each slot of the next cycle Do
18:    If the node receives the RRM then
19:      For in accordance with the order of reception for each RRM Do
20:        If $k_f + 1 < k_c$ then
21:          $k_c = k_f + 1$
22:          $C_t = \sum C_n{}'$
23:          If routing already exists then
24:            Broadcast DRM
25:          End if
26:          Broadcasts CRM
27:        End if
28:        If $k_f + 1 = k_c$ and $\sum C_n{}' < C_t$ then
29:          $k_c = k_f + 1$
30:          $C_t = \sum C_n{}'$
31:          If routing already exists then
32:            Broadcast DRM
33:          End if
34:          Broadcast CRM
35:        End if
36:      End for
37:    End if
38:End for
39:End Loop
40:$k' = k' + 1$
41:If There are nodes with $k = \infty$ in n nodes
42:  Goto Loop
```

The routing algorithm has been described above. On the basis of the network adjusted by asynchronous slot, the routing algorithm is demonstrated in the $v_1 \sim v_{15}$ region shown in Fig. 11 *(note: please ignore the gray color of the notes)*.

Table 8 shows the details of each node from v_1 to v_{15}.

Table 7 Adjusted table-a

Notation	Description
id	The id of the current node
flag	Binary-state variable, the flag that the node active slot is asynchronous in the sensing range
$flag'$	Binary-state variable, the flag that the node active slot is asynchronous in the transmission range
φ_c	The active slot of the current node
ξ	The number of neighbors of the current node (including itself)
r	A collection containing repeated active slots in the sensing range
k	The cost from the current node to the sink node
$\overrightarrow{v}$	$\overrightarrow{v} = (v_0, v_2, v_3 \cdots v_M)$, row vector, v_i. The value of v_i is binary state, and v_i=1 means that a node whose active slot is i exists in the current sensing range.
C_n	The current node n_c is connected to the number of nodesn_t whose hop count is greater than the number of hops. The initial value is 0.
C_t	The current node n_c is connected to the number of nodes n_t whose hop count is less than the number of hops. The initial value is ∞.

Initially, in the first cycle, each node exchanges data with the neighbors of $d' \le R_t$ and initializes its own Table 1 and Table 2. In the last time slot of the first cycle, that is, slot = 9, the sink node sends a route request message (RRM). As shown in Fig. 11 *(note: please see the entire whole figure)*, only v_1, v_2, and v_3 can receive this message.

In the second cycle, the three nodes of $k = \infty$,v_1, v_2, and v_3 receive the RRM from sink in slots 1, 2, and 6 in turn. Update k to 1, C_t to sink's C_n. The three nodes establish routing with sink respectively. The updated information of ϕ_c, hop, and so on is shown in Table 9. In the third cycle, three nodes v_1,v_2, and v_3 with $k = k' = 1$ send RRMs when they are active. The RRM reachable nodes of the three nodes are shown in Table 10. From the table, we can see that some nodes have received more than one RRM, so it is necessary to select a better node to establish a routing according to the established routing situation of the target node.

From the table, we can see that some nodes have received more than one RRM, so it is necessary to select a better node to establish a routing according to the established routing situation of the target node.

In the fourth cycle, when slot = 4, v_4 receives RRM from v_1 and v_2 and processes RRM according to the receiving order. Assuming that the closer the distance between two nodes is, the earlier RRM arrives, v_4 takes the lead in processing RRM from v_1.Because $k_f + 1 < k_c$, v_4, and v_1 establish routing and $k_c = 1$, $C_t = 0$ for v_4 broadcast CRM. Processing RRM from v_2, $\Sigma C_n{}' = C_t$, and $k_f + 1 = k_c$ does not satisfy the conditions of the algorithm, does not establish a routing, and ends the two RRM processing.

Slot = 5, v_6 revives RRM from RRM, because $k_f + 1 < k_c$, v_6, establishes a route with v_2, $k_c = 1$, $C_t = 0$ for v_6, broadcast CRM.

When slot = 7, v_5 receives RRM from v_1 and CRM from v_4 broadcasting, which contains the message of C_n plus 1 of v_1, so $\Sigma C_n{}' = 1$, $C_t = \infty$ of v_5 satisfies $\Sigma C_n{}' < C_t$, v_5 establishes routing with v_1, $k_c = 1$, $C_t = 1$ for v_5, broadcasting CRM.

When slot = 9, v_7 receives three RRMs from v_1,v_2, and v_3, and CRM broadcasted by v_6. The RRM is processed in the order of reception, and the processing order is v_3,v_2, and v_1. At this time, the details of the three nodes are as Table 11 shows. Based on this information, v_7 will decide which node to establish the route with.

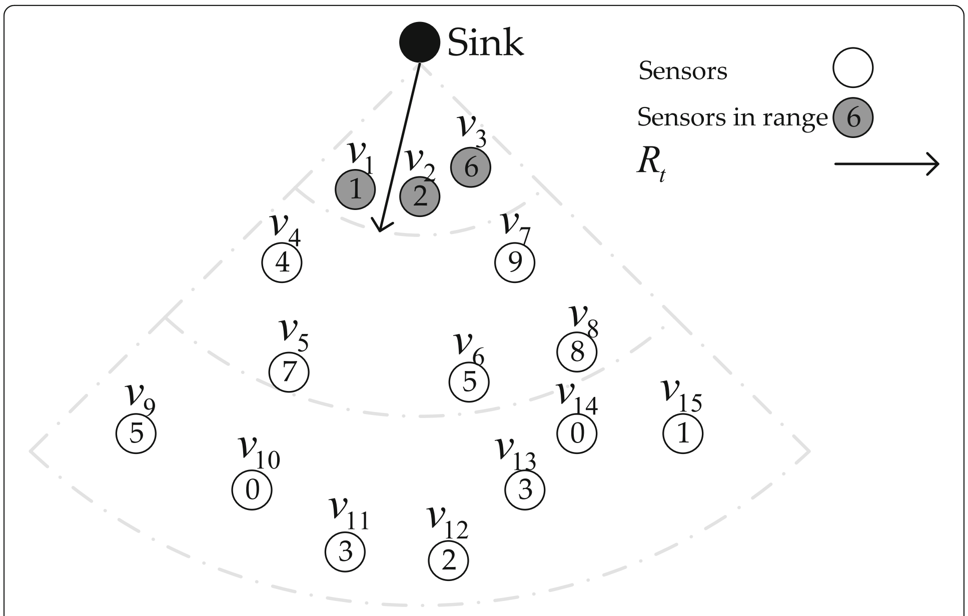

Fig. 11 v_1~v_15 region (note: please ignore the gray color of the notes) \ and \ Sink broadcast RRM (note: please see the entire whole figure)

v_7 first establishes a route with v_3, $k_c = 1$, at which time the C_t of v_7 is 0.

When v_7 processes the RRM from v_2, the C_n of v_2 is 1, which is greater than the C_t of v_7, and no route is established. Afterwards continuing to process the RRM from v_1, since the requirement of the algorithm is not satisfied, none of route is established. After the end of the four cycles, the established route is shown in Fig. 12 *(note: please ignore the route lines in the sector between the second arc and the third arc).*

In the fifth cycle, four nodes v_4,v_5,v_6, and v_7, satisfying $k = k' = 2$, are sent to RRM at active time respectively. The nodes that can receive RRMs sent by previous four nodes and satisfy the condition of $k = \infty$ are shown in Table 12.

Table 8 The details of each node from v_1 to v_{15}

	φ_c	k	C_n	C_t
v_1	1	∞	0	∞
v_2	2	∞	0	∞
v_3	6	∞	0	∞
v_4	4	∞	0	∞
v_5	7	∞	0	∞
v_6	5	∞	0	∞
v_7	9	∞	0	∞
v_8	8	∞	0	∞
v_9	5	∞	0	∞
v_{10}	0	∞	0	∞
v_{11}	3	∞	0	∞
v_{12}	2	∞	0	∞
v_{13}	3	∞	0	∞
v_{14}	0	∞	0	∞
v_{15}	1	∞	0	∞

Table 9 The updated information of v_1, v_2, and v_3

	φ_c	k	C_n	C_t
v_1	1	1	0	0
v_2	2	1	0	1
v_3	6	1	0	2

As can be referred from Table 12, five nodes v_9, v_{10}, v_{11}, v_{12}, v_{13} receive only one RRM, so a route is established with the node that sends the RRM.

In the sixth cycle, when the slot is 0, v_{10}, v_{14} establish routes with v_5, v_6, respectively, and the hops are updated to 3.

When slot is 2, v_{12} establishes a route with v_6, and the hop is updated to 3.

When slot is 3, v_{11}, v_{13} establish routes with v_5, v_6, respectively, and the hops are updated to 3.

When slot is 5, v_9 establishes a route with v_5, and the hop is updated to 3.

When slot is 6, since v_{12}, v_{13}, v_{14} and v_6 have established routes before 8 slot, v_8 will establish a route with RRM from v_6, but when dealing with the routing information from v_7, it will broadcast DRM first, then CRM, and finally establish a routes with v_7, and hop is updated to 3.

After six cycles, the established route is shown in Fig. 12 *(Note: please see the entire whole figure)*.

In the seventh cycle, the six nodes with $k = k^{'} = 3$ send RRMs when they are active. In the scope of our demonstration, only the hop of v_15 is ∞.It can receive RRMs from v_6, v_8, v_{13}, v_{14}.In the eighth cycle, when the slot 1, v_{15} receives four RRMs. According to the distance, the processing order is v_{14}, v_8, v_{13}, v_6. According to the constraints in the algorithm, v_{14} is finally established with v_{15}.

After eight cycles, v_1~v_{15} can transmit information with the sink. Finally, the information about these 15 nodes is shown in Table 13.

After a number of cycles, the entire network has established routing, and the entire network topology is shown in Fig. 13.

4.2.3 Continuous adjustment algorithm

After algorithm 1 and algorithm 2, the nodes in the network have ensured that they are asynchronous within the sensing range, and a route suitable for continuous slot adjustment is established. In order to reduce the delay caused by the message transmission, a time slot continuous adjustment algorithm is designed.

The main process is as follows:

Initially, multiple nodes with the largest hop send a request message for slot adjustment to the nodes within the transmission range. Nodes with smaller hops receive a new active slot based on the data of the message. When the calculated active slot

Table 10 The nodes that RRMs for v_1, v_2, v_3 can reach

node	Reachable node
v_1	v_4, v_5, v_7
v_2	v_4, v_6, v_7
v_3	v_7

Table 11 Information stored in v_1, v_2, and v_3 when slot = 9

	φ_c	k	C_n	C_t
v_1	1	1	0	0
v_2	2	1	0 + 1	1
v_3	6	1	0	2

conflicts with the asynchronous condition required by algorithm 1, the average of two active slot is used as the node's new active slot. In the process of adjustment, two tables of itself should be updated, and slot update messages should be sent to nodes within the transmission range.

In algorithm 3, we have the following conventions:

1. At the beginning of algorithm 3, the active slots of each node in the network are adjusted by algorithm 1 and routed by algorithm 2.
2. The continuous time slot adjustment request message (CSAM) includes the adjustment request, the node's own id, and active slot (ϕ_c). For convenience, when a successor node receives multiple CSAMs, ϕ_{ci}' is used to denote ϕ_c in the ith message.
3. After each adjustment of the node's active slot, the following actions are taken:

a) Sending a slot update message to neighbors those satisfy the condition of $d \leq R_t$
b) Update table-a and table-b

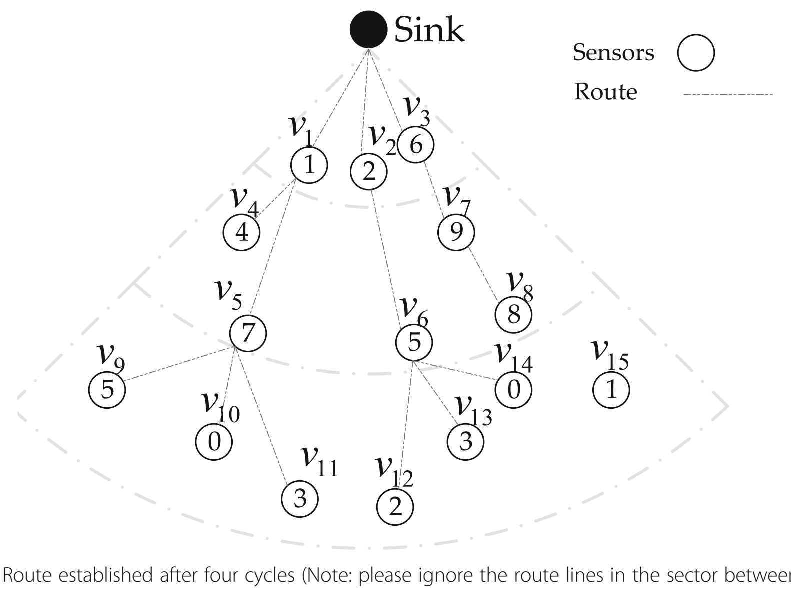

Fig. 12 Route established after four cycles (Note: please ignore the route lines in the sector between the second arc and the third arc), \ and \Route established after six cycles (note: please see the entire whole figure)

```
Algorithm 3:Continuous adjustment algorithm
1:$k' = 0$
2:For each slot in the current cycle Do
3:  If $k' < k_c$ then
4:    $k' = k_c$
5:  End if
6:End for
7:Loop:
8:For each slot in the next cycle Do
9:    If $k_c = k'$ then
10:     Send CSAM to routing successor node
11:     flag1 = 1
12:   End if
13:End for
14:For each slot in the next cycleDo
15:   If $k_c = k' - 1$ then
16:     Receive the CSAM sent by the predecessor node (the number is greater than or equal to 0)
17:     If there is $\varphi_t$ that satisfies the following conditions:
          $\sum_{i=1}^{C_n}(\varphi_t - \varphi_{ci}' + m) mod(m+1)$ reaches the minimum value (non-negative) and flag = 1 then
18:       $\varphi_c = \varphi_t$
19:       flag1 = 1
20:     End if
21:     Else if the calculated $\varphi_t$ cannot satisfy flag = 1 then
22:       $\varphi_c = \varphi_t + 1$
23:       flag1 = 1
24:     End else if
25:   End if
26:End for
27:End Loop
28:$k' = k' - 1$
29:If $k' \neq 0$ then
30:  Goto Loop
31: End if
```

The slot continuously adjustment algorithm has been described above.

Based on the Fig. 13 network after the asynchronous slot adjustment and routing establishment, the following example is shown in Fig. 14 as an example to demonstrate the continuous slot adjustment algorithm.

The number in the lower right corner of the node in the figure represents the k of the current node.

In the first cycle, get the maximum k in the current network and get the initial $k' = 8$.

Table 12 The nodes that RRMs forv_4, v_5, v_6, v_7 can reach

Node	Reachable node
v_4	/
v_5	v_9, v_{10}, v_{11}
v_6	v_8, v_{12}, v_{13}, v_{14}
v_7	v_8, v_{14}

Table 13 State of each node after eight cycles

	φ_c	k	C_n	C_t
v_1	1	1	2	0
v_2	2	1	1	0
v_3	6	1	1	0
v_4	4	2	0	0
v_5	7	2	3	1
v_6	5	2	3	0
v_7	9	2	1	0
v_8	8	3	0	0
v_9	5	3	0	2
v_{10}	0	3	0	0
v_{11}	3	3	0	1
v_{12}	2	3	0	1
v_{13}	3	3	0	2
v_{14}	0	3	0	0
v_{15}	1	4	0	0

In the second cycle, three nodes satisfying $k = k^{'} = 8$, namely, v_{59}, v_{60}, v_{61} send CSAM to their respective successors.

In the third cycle, two nodes v_{43}, v_{44} satisfying $k = 7$ and receiving the CSAM process the received message and calculate ϕ_t which makes $\sum_{i=1}^{C_n}(\phi_t - \phi_{ci}{'} + m)\ mod(\ m+1)$ to the minimum value. The two results are 0 and 7, and the constraints of algorithm 1

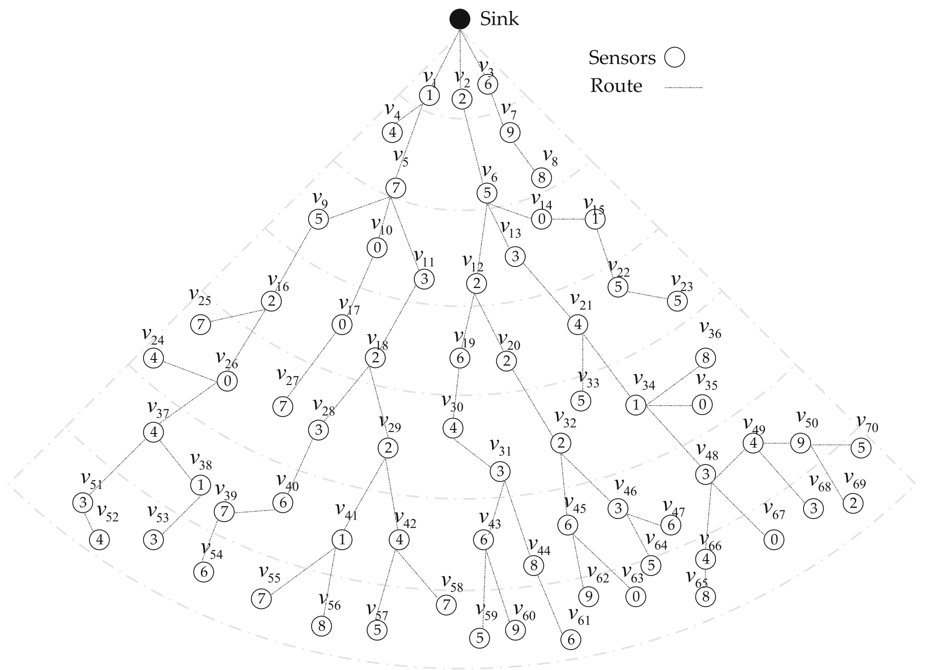

Fig. 13 The WSNs after algorithm 1 and algorithm **2**

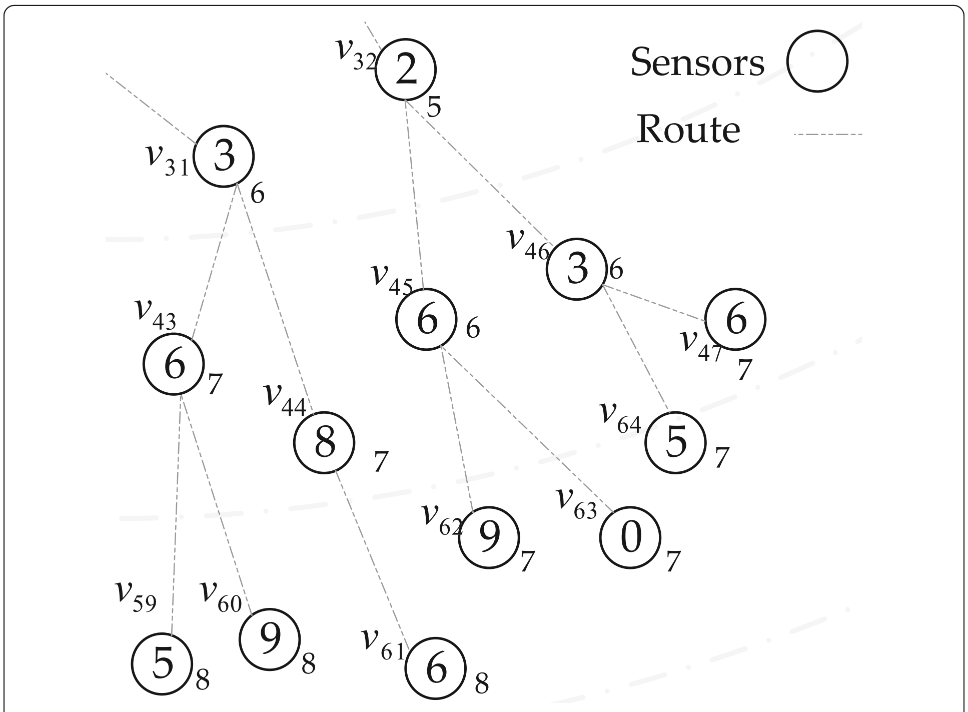

Fig. 14 Demonstrate the area of the slot continuous adjustment algorithm

can be guaranteed. Therefore, the active slots of v_{43} and v_{44} are updated to 0 and 7. At this time, part of the network status is shown in Fig. 15 *(note: please ignore the brackets and the numbers in brackets)*.

In the fourth cycle, six nodes, satisfying $k = k' = 7$ condition, namely, v_{43}, v_{44}, v_{62}, v_{63}, v_{64}, v_{47} send CSAM to their respective successors.

In the fifth cycle, three nodes v_{31}, v_{45}, v_{46} satisfying $k = 6$ and receiving the CSAM condition process the received message and calculate ϕ_t which makes $\sum_{i=1}^{C_n}(\phi_t - \phi_{ci}' + m)$ $mod(\ m+1)$ to the minimum value. The three results are 1, 1, and 7, and the constraints of algorithm 1 can be guaranteed. Therefore, the active slots of v_{31}, v_{45} and v_{46} are updated to 1, 1, and 7. At this time, part of the network status is shown in Fig. 15 *(note: for the node with bracket in the figure, the number of the node is the number in the brackets)*.

In the sixth cycle, two nodes satisfying $k = k' = 8$ condition, namely, v_{45}, v_{46} send CSAM to their respective successors. In the seventh cycle, the node v_{32} satisfying $k = 5$ and receiving the CSAM processes the received message and calculates ϕ_t which makes $\sum_{i=1}^{C_n}(\phi_t - \phi_{ci}' + m)\ mod(\ m+1)$ to the minimum value. The result is 2, and v_{32} does not need updating.

After 7 cycles, all nodes in this part of the network are adjusted continuously.

After several cycles, all nodes in the network are adjusted continuously, and the network is shown in Fig. 16.

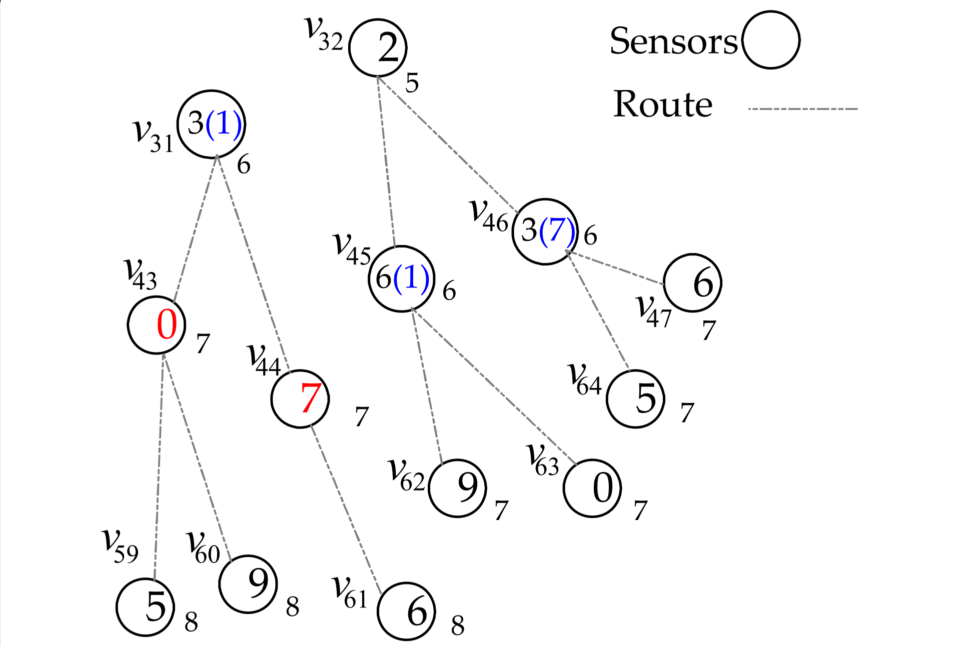

Fig. 15 v_{43}, v_{44} in wireless sensor networks are continuously adjusted *(note: please ignore the brackets and the numbers in brackets)*, and v_31, v_45, and v_46 in wireless sensor networks are continuously adjusted *(note: for the node with bracket in the figure, the number of the node is the number in the brackets)*

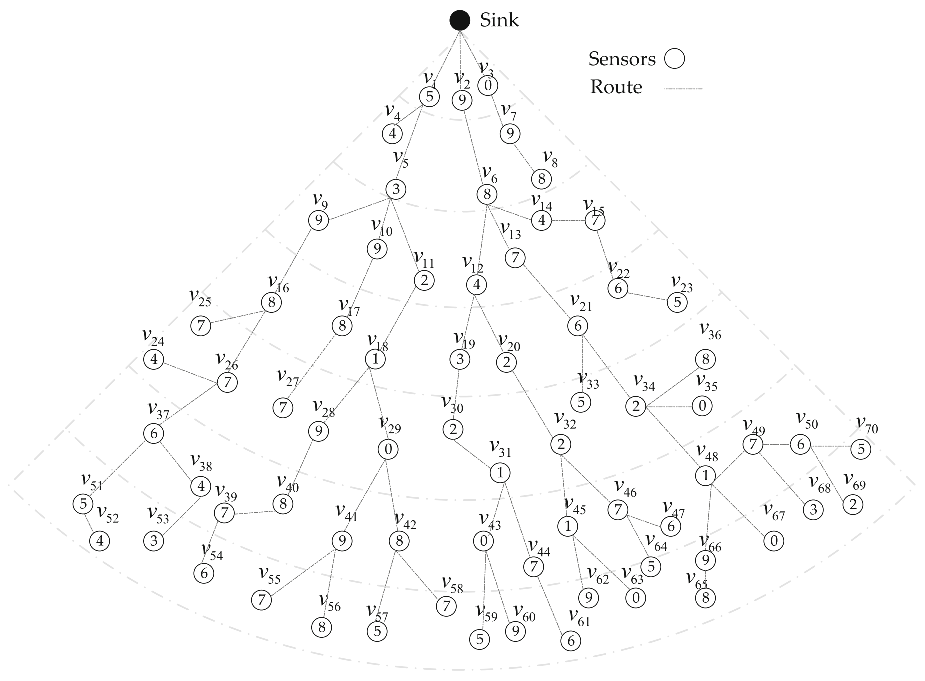

Fig. 16 Bi-adjusted and routed WSNs

So far, the adjustment of the initial network has been completed, and a wireless sensor networks with asynchronous active slot in the sensing range, continuous active slot in the transmission range, and uniform routing has been obtained.

In order to compare the event detection latency, data routing delay and total delay on the initial network and the adjusted network, three fire points are placed at the same location on both networks. It is stipulated that the three fire points are consistent, and the fire slots in one cycle is 1, 5 and 8. After the fire point is arranged, the fire is simulated and the fire occurs in different slots. Fig. 17 shows the sensor of the sensed event and the path of the data packet routing in the case of different fire slots on the initial network. As can be seen from the figure, in the current network state, no matter which slot of 1, 5, 8 fire occurs, the sending sensors and path are the same.

According to the above simulation situation, the delay of different positions and different fire occurrence slots are calculated and shown in Table 14.

The average detection latency ($\overline{S}$), the average data routing delay ($\overline{T}$), and average total delay ($\overline{D}$) in the initial network and Bi-adjusted network are calculated according to Table 14, and the percentage of delay reduction after bi-adjusting is calculated as shown in Table 15.

From the table, we can see that the average event detection latency decreases by 35.80%, the average data routing delay decreases by 45.13%, and the total delay decreases by 43.62%.

This shows that in the current network situation, the Bi-adjusting has an effect on the delay reduction, and the effect is more obvious in reducing the data routing delay.

The main work of this part is to propose slot adjustment algorithms and routing algorithm. After each algorithm was described, it was demonstrated on a network of 70

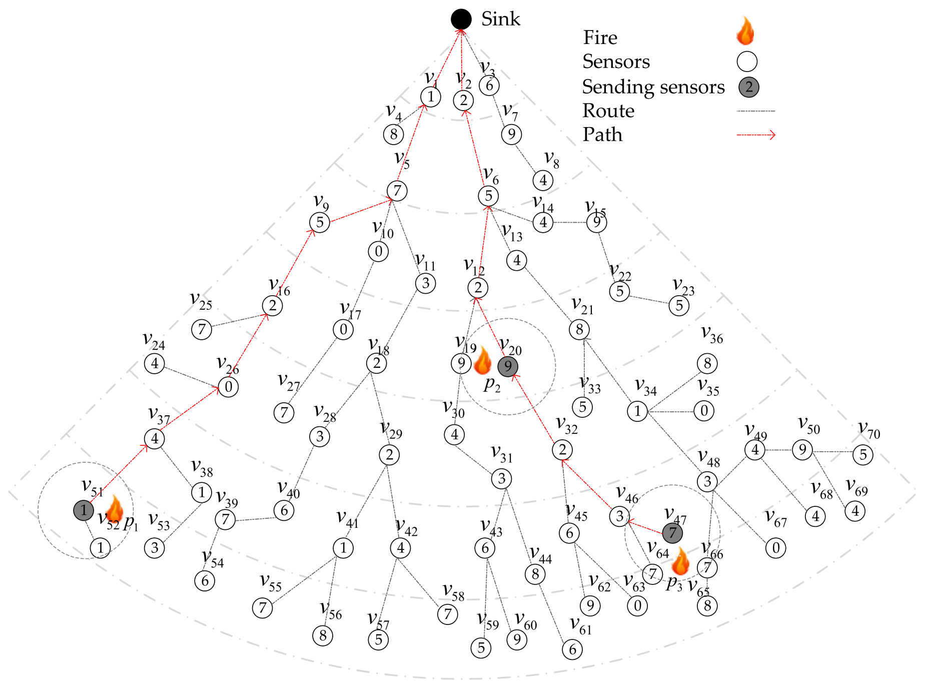

Fig. 17 Fire slot = 1, 5, 8 sending sensors and path on the initial network

Table 14 Delay calculation result

	Fire slot	Random delay	bi-adjusting delay
p_1	1	0 + 20 = 20	3 + 11 = 14
	5	6 + 20 = 36	0 + 10 = 10
	8	3 + 20 = 23	6 + 11 = 17
p_2	1	8 + 13 = 21	1 + 7 = 8
	5	4 + 13 = 17	7 + 7 = 14
	8	1 + 13 = 14	4 + 7 = 11
p_3	1	6 + 35 = 41	4 + 24 = 28
	5	2 + 35 = 37	0 + 24 = 24
	8	9 + 35 = 44	0 + 11 = 11

nodes. Finally, the fire is simulated on the adjusted network and the initial network. The number of delayed slots is calculated, and the results are analyzed. It is found that BADCS can effectively reduce delay.

5 The theoretical analysis

The above analysis is based on the specific example, and in this part, we will theoretically analyze the effect of BADCS to reduce the event detection latency and data routing delay.

We stipulate that there are n sensor nodes in the perception range centered on the fire point, and m slots in a cycle, with slot numbers ranging from 0 to m-1.

In the case of randomly arranged node slots and randomly (0~m-1) fired slots, the formula for calculating event detection latency is $S = \sum_{i=0}^{m-1} i \cdot P_i$. In the formula, i represents the number of delayed slots, and P_i represents the probability of delaying i slots.

Assuming that the fire event occurs at 0 slot, to make the event detection latency zero, it is necessary to ensure that at least one of the n nodes wakes up at 0 slot that is:

$$P_0 = \frac{m^n - (m-1)^n}{m^n} \tag{6}$$

To make event detection latency be 1 slot, it is necessary to ensure that no node wakes up at slot 0, and at least one node wakes up at slot 1 that is:

$$P_1 = \frac{(m-1)^n - (m-2)^n}{m^n} \tag{7}$$

To make event detection latency be 2 slots, it is necessary to ensure that no node wakes up at slot 0 and 1, and at least one node wakes up at slot 2 that is:

Table 15 Average delay and percentage reduction

	Random slot	Bi-adjusting slot	Percent decrease
$\overline{S}$	4.33	2.78	35.80%
$\overline{T}$	22.67	12.44	45.13%
$\overline{D}$	27	15.22	43.62%

$$P_2 = \frac{(m-2)^n - (m-3)^n}{m^n} \tag{8}$$

Then, we get the general formula:

$$P_i = \frac{(m-i)^n - (m-i-1)^n}{m^n} (i = 0, 1, 2 \cdots m-1) \tag{9}$$

Regardless of the slot in which the fire occurs, the event detection latency can be calculated using the general formula. Therefore, if the fire occurrence time slot is a random slot in 0 ~ m-1, the average event detection latency is:

$$\overline{\mathrm{S}} = \sum_{i=0}^{m-1} i \cdot P_i \tag{10}$$

When the node time slot is asynchronously adjusted, the slots of the n sensor nodes are asynchronous in the sensing range centered on the fire point. In this case, if $n \geq m$, meaning that the number of nodes is greater or equal to the length of the period, the event detection latency is 0. If $n < m$, n node slot arrangement situation has A_m^n kinds of possibilities and $A_m^n = \frac{m!}{(m-n)!}$, so $P_i = A_{m-i}^n - A_{m-i-1}^n (i = 0, 1, 2 \cdots m-n-1)$

Therefore, the average event detection latency calculation formula after asynchronous adjustment is as follows:

$$\overline{\mathrm{S}} = \begin{cases} 0, n \geq m \\ \sum_{i=0}^{m-1} i \cdot P_i, n < m \end{cases} \tag{11}$$

For data routing delay, we assume that there are n nodes (excluding sink) on a route, and there are m slots in one cycle in which the slot number is from 0 to m-1.

In the case of randomly arranging the node active slot, we give the formula for the data routing delay of a route where the information is sent from the node with the largest hop to the sink node:

$$\mathrm{T} = \frac{1 + 2 + \cdots + m}{m^2} m \cdot (n-1) \tag{12}$$

After simplifying, we get:

$$\mathrm{T} = \frac{1+m}{2} \cdot (n-1) \tag{13}$$

After the continuous adjustment, the optimal situation is reached, and the data routing delay is $n - 1$.

According to the theoretical analysis, when the slots number m and the nodes number n are assigned different values, P_i, event detection latency, data routing delay, and reduced delay can be observed.

It can be seen from Table 16 that a special case appears when n=1. And in this case, if there is only one wireless sensor node in the range of the sense of the fire point, then the probability of EDL for any number of slots is 5%. In the general case (n≠1), when the number of nodes is the same, the probability that the number of delayed slots is less corresponding to a larger probability. As the number of delay slots increases, the probability gradually decreases, until to 0.

Table 16 The EDL probability variation under the condition of randomly arranging the sensor node active slot, when m = 20 and n = 1, 10, 20, 30

Slots of delay	n = 1 (randomly)	n = 10 (randomly)	n = 20 (randomly)	n = 30 (randomly)
0	5	40.1263	64.1514	78.5361
1	5	25.0058	23.6909	17.2248
2	5	15.1804	8.28171	3.47604
3	5	8.95002	2.72303	0.639282
4	5	5.10607	0.8358	0.105936
5	5	2.8066	0.237329	0.0156043
6	5	1.47848	0.0616677	0.00200993
7	5	0.741613	0.0144684	2.22E-04
8	5	0.351367	0.00301457	2.05E-05
9	5	0.155639	5.46E-04	1.53E-06
10	5	0.0636056	8.38E-05	8.92E-08
11	5	0.0235649	1.05E-05	3.83E-09
12	5	0.00772721	1.02E-06	1.13E-10
13	5	0.00216806	7.26E-08	2.08E-12
14	5	4.95E-04	3.40E-09	2.05E-14
15	5	8.51E-05	8.99E-11	8.66E-17
16	5	9.66E-06	1.05E-12	1.07E-19
17	5	5.67E-07	3.32E-15	1.92E-23
18	5	9.99E-09	1.00E-18	1.00E-28
19	5	9.77E-12	9.54E-25	9.31E-38

Overall, the more nodes are, the greater the probability of obtaining a smaller value such as delay = 0, 1 is, which means that when the sensor node active slot is randomly arranged, the increase in the number of nodes is beneficial to the increase in low EDL probability.

Next, we will observe whether the above conclusion is established after asynchronous adjustment.

As can be seen from Table 17, when $n = m$, the probability percentage of EDL being zero is 100%, that is, no EDL. Like Table 16, when n=1, regardless of the slot, the delay probability is 5%. Observe the curve under normal conditions and find that the above conclusion is true. If the above conclusion is true, then according to the formula

$$\overline{\mathrm{S}} = \sum\nolimits_{i=0}^{m-1} i{\cdot}P_i, \text{and } \overline{\mathrm{S}} = \begin{cases} 0, n \geq m \\ \sum\nolimits_{i=0}^{m-1} i{\cdot}P_i, n < m \end{cases} \tag{14}$$

The average EDL should decrease as the number of nodes increases.

It can be seen from Table 18 that when m is a fixed value, with the increase of n, the average EDL of randomly arranged active slots and asynchronously adjusted networks decreases gradually, and the average EDL of asynchronously adjusted networks is always less than that of the initial networks. The previous analysis is based on the case where the number of time slots is constant, and the number of nodes varies. Next, the

Table 17 The EDL probability variation under the condition of asynchronously arranging the sensor node active slot, when m = 20 and n = 1, 5, 10, 15, 20

Slots of delay	n = 1 adjusted	n = 5 adjusted	n = 10 adjusted	n = 15 adjusted	n = 20 adjusted
0	5	25	50	75	100
1	5	19.7368	26.3158	19.7368	0
2	5	15.3509	13.1579	4.38596	0
3	5	11.7389	6.19195	0.773994	0
4	5	8.80418	2.70898	0.0967492	0
5	5	6.4564	1.08359	0.00644995	0
6	5	4.61171	0.386997	0	0
7	5	3.19272	0.119076	0	0
8	5	2.12848	0.029769	0	0
9	5	1.35449	0.0054125	0	0
10	5	0.812693	0	0	0
11	5	0.451496	0	0	0
12	5	0.225748	0	0	0
13	5	0.0967492	0	0	0
14	5	0.0322497	0	0	0
15	5	0.00644995	0	0	0
16	5	0	0	0	0
17	5	0	0	0	0
18	5	0	0	0	0
19	5	0	0	0	0

case where the number of nodes n is fixed at 10 and the number of slots m is constantly changing will be discussed.

From Table 19 and Table 20, we can see that the probability of low event detection latency gradually decreases with the increase of slots in the cycle within the sensing range with the fire point as the center of the circle, regardless of whether the active slots are adjusted asynchronously or not.

As can be seen from Table 21, when n is a fixed value, as m increases, the average EDL of the randomly arranged active slot and the network adjusted asynchronously gradually increases. In order to observe the effect of reducing the average EDL by the asynchronous adjustment of the nodes further, Table 22 and Table 23 are drawn.

As can be seen from Table 22, in the case where n is fixed, taking m=15 and m=25 as an example, as m increases, the probability of less delay decreases, and the probability of larger delay increases. For the asynchronous adjustment, it can be seen from the figure that in the case where both m and n are the same, after the adjustment, the probability of less delay is significantly increased, which is beneficial to reducing the average EDL.

As can be seen from Table 23, in the case where m is fixed, taking n = 10 and n=15 as an example, as m increases, the probability of event with less delay increases, while the probability of larger delay decreases. For the asynchronous adjustment, it can be seen from the figure that in the case where m and n are the same, the probability of event with less delay is significantly increased after adjusting, which is beneficial to reducing average EDL.

Table 18 The average EDL delay slots under the condition of randomly/asynchronously arranging the sensor node active slot, when m = 20, and n increased from 1 to 50

n	1	2	3	4	5	6
Randomly	9.5	6.175	4.5125	3.51666	2.85416	2.38212
Adjusted	9.5	6	4.25	3.2	2.5	2
7	8	9	10	11	12	13
2.02913	1.7555	1.53741	1.35972	1.21233	1.08823	0.982442
1.625	1.33333	1.1	0.909091	0.75	0.615385	0.5
14	15	16	17	18	19	20
0.89129	0.81203	0.742559	0.681244	0.626792	0.578172	0.534546
0.4	0.3125	0.235294	0.166667	0.105263	0.05	0
21	22	23	24	25	26	27
0.49523	0.45966	0.427363	0.397943	0.371066	0.346445	0.323836
0	0	0	0	0	0	0
28	29	30	31	32	33	34
0.303027	0.283836	0.266102	0.249687	0.234466	0.220333	0.20719
0	0	0	0	0	0	0
35	36	37	38	39	40	41
0.194953	0.183545	0.172897	0.16295	0.153647	0.144939	0.136781
0	0	0	0	0	0	0
42	43	44	45	46	47	48
0.129131	0.121954	0.115214	0.108881	0.102927	0.0973254	0.092053
0	0	0	0	0	0	0
49	50					
0.0870877	0.0824094					
0	0					

In Table 22 and Table 23, we have shown the average EDL in random active slot and asynchronous active slot. By subtracting the EDL after asynchronous adjustment from the EDL at random, we can get the reduced delay through asynchronous adjustment and get Table 24.

From Table 24, we can see that each curve has a maximum value in the existing range, and the number of slots m is equal to the number of nodes n. When $m = n$, the EDL that can be reduced by asynchronous adjustment is the largest.

When $m < n$, with the increase of m, the reduction of EDL is also increasing, but when $m > n$, with the increase of m, the reduction of EDL is decreasing. To demonstrate the role of asynchronous adjustment in reducing EDL, Table 25 shows the percentage of EDL reduced by asynchronous adjustment in the case of random different number of nodes and slots.

As can be seen from the figures, when $n \le m$, that is, when the number of sensor nodes is not more than the number of slots, the percentage of EDL reduced by asynchronous adjustment is 100%. Because in this case, the event detection latency adjusted asynchronously is always 0. With the increase of m, the percentage decreases gradually, which indicates that the more the number of slots is, the closer random layout is to asynchronous layout.

Table 19 The low EDL probability variation under the condition of randomly arranging the sensor node active slot, when n = 10, m = 10, 15, 20, 25

Slots of delay	n = 10			
	m = 10	*m* = 15	*m* = 20	*m* = 25
0	65.1322	49.8388	40.1263	33.5167
1	24.1304	26.2544	25.0058	23.0444
2	7.91267	13.1694	15.1804	15.5887
3	2.22009	6.23947	8.95002	10.36
4	0.507006	2.76379	5.10607	6.7527
5	0.0871705	1.12949	2.8066	4.30853
6	0.0098953	0.418459	1.47848	2.68498
7	5.80E-04	0.137218	0.741613	1.62998
8	1.02E-05	0.0384998	0.351367	0.961001
9	1.00E-08	0.0087923	0.155639	0.54826
10	–	0.0015117	0.0636056	0.301356
11	–	1.72E-04	0.0235649	0.15875
12	–	1.01E-05	0.0077272	0.07963
13	–	1.77E-07	0.0021681	0.0377277
14	–	1.73E-10	4.95E-04	0.0167116
15	–	–	8.51E-05	0.0068296
16	–	–	9.66E-06	0.0025303
17	–	–	5.67E-07	8.30E-04
18	–	–	9.99E-09	2.33E-04
19	–	–	9.77E-12	5.32E-05
20	–	–	–	9.14E-06
21	–	–	–	1.04E-06
22	–	–	–	6.08E-08
23	–	–	–	1.07E-09
24	–	–	–	1.05E-12

As for DRD, the following analysis is carried out under the optimum conditions because it is more convenient to calculate the optimum conditions after successive adjustment.

It can be seen from Table 26 that when the number of slots m is constant, DRD increases with the increasing number of nodes in a routing, and when the number of nodes n is constant, DRD increases linearly with the increasing m.

As can be seen from Table 27, the reduction in DRD after continuous adjustment is similar to its change trend. When the number of time slots m is constant, as the number of nodes n on a route increases, the reduction of DRD increases. When the number of nodes is constant, the decrease of DRD increases linearly with the increase of slot number m.This is because the reduction is calculated according to the optimal condition after the subsequent adjustment, and the reduction is all $n-1$ here.

As a distributed algorithm, we briefly introduce the message complexity of methods. In WSNs, when a node's active slot needs to be adjusted, the nodes communicate with each other. Within the sensing range, it is assumed that there are n nodes, and only one message is sent in each communication. Considering the worst case, the active

Table 20 The low EDL probability variation under the condition of asynchronously arranging the sensor node active slot, when n = 10 and m = 10, 15, 20, 25

Slots of delay	n = 10			
	m = 10	m = 15	m = 20	m = 25
0	100	66.6667	50	40
1	0	23.8095	26.3158	25
2	0	7.32601	13.1579	15.2174
3	0	1.8315	6.19195	8.99209
4	0	0.333	2.70898	5.13834
5	0	0.0333	1.08359	2.82609
6	0	0	0.386997	1.48741
7	0	0	0.119076	0.743707
8	0	0	0.029769	0.34998
9	0	0	0.0054125	0.153116
10	N/A	0	5.41E-04	0.0612465
11	N/A	0	0	0.0218737
12	N/A	0	0	0.0067304
13	N/A	0	0	0.0016826
14	N/A	0	0	3.06E-04
15	N/A	N/A	0	3.06E-05
16	N/A	N/A	0	0
17	N/A	N/A	0	0
18	N/A	N/A	0	0
19	N/A	N/A	0	0
20	N/A	N/A	N/A	0
21	N/A	N/A	N/A	0
22	N/A	N/A	N/A	0
23	N/A	N/A	N/A	0
24	N/A	N/A	N/A	0

slots of all nodes are the same, and each node needs to communicate with the rest of the neighborhood; then, $n \times (n - 1)$ communications are required, and the message complexity can be expressed as $O(n^2)$. In a path, if there are n nodes and only one message is sent for each communication, each node needs to send message forward $(n - 1)$, times of communication is required, and the message complexity can be expressed as $O(n)$.

6 Experimental and results

In the experiment, we used a randomly generated WSNs to compare the performance of the BADCS scheme network with the initial network in terms of latency. The parameter settings of the WSNs used in the experiment are shown in Table 28.

Based on the set parameters, a wireless sensor network as shown in Fig. 4 is formed. Under the initial conditions, the active slots of the 90 sensor nodes on the WSNs are randomly set, and the details are shown in Table 29.

Firstly, algorithm 1 is executed, and each node exchanges data with the neighbors. In this step, the nodes of v_9 , v_{16}, v_{18}, v_{22}, v_{23}, v_{26}, v_{27}, v_{28}, v_{29}, v_{37}, v_{38}, v_{62}, v_{69}, v_{70}, v_{82}, v_{84},

Table 21 The average EDL delay slots under the condition of randomly/asynchronously arranging the sensor node active slot, when *n* = 20, and *m* increased from 1 to 50

m	1	2	3	4	5	6
Randomly	0	9.77E-04	0.0173585	0.057291	0.113526	0.179841
Adjusted	0	0	0	0	0	0
7	8	9	10	11	12	13
0.252555	0.329517	0.409419	0.491434	0.575012	0.659779	0.745468
0	0	0	0	0.0909091	0.181818	0.272727
14	15	16	17	18	19	20
0.831889	0.918897	1.00639	1.09427	1.18249	1.27099	1.35972
0.363636	0.454545	0.545455	0.636364	0.727273	0.818182	0.909091
21	22	23	24	25	26	27
1.44867	1.53779	1.62706	1.71647	1.806	1.89563	1.98536
1	1.09091	1.18182	1.27273	1.36364	1.45455	1.54545
28	29	30	31	32	33	34
2.07517	2.16506	2.25501	2.34503	2.4351	2.52522	2.61539
1.63636	1.72727	1.81818	1.90909	2	2.09091	2.18182
35	36	37	38	39	40	41
2.7056	2.79585	2.88614	2.97646	3.06681	3.15718	3.24758
2.27273	2.36364	2.45455	2.54545	2.63636	2.72727	2.81818
42	43	44	45	46	47	48
3.33801	3.42846	3.51893	3.60942	3.69992	3.79045	3.88099
2.90909	3	3.09091	3.18182	3.27273	3.36364	3.45455
49	50					
3.97154	4.06211					
3.54545	3.63636					

v_{88}, v_{89} that do not satisfy the asynchronous condition are found, and their flag value is adjusted to 0. And at the same tine, the corresponding wireless sensor network is modified synchronously. After the asynchronous adjustment, the active slot of the sensing node in the WSNs changes as shown in Table 30.

Secondly, apply algorithm 2 to establish routes based on the initial WSNs network, and the WSNs with routes being set up is shown in Fig. 18. And we also apply

Table 22 Comparison of the probability of different delay at *n* = 10, different *m*, and different active slot setting strategies

Slots of delay	*m* = 15	*m* = 25	*m* = 15, adjusted	*m* = 25, adjusted
0	49.8388	33.5167	66.6667	40
1	26.2544	23.0444	23.8095	25
2	13.1694	15.5887	7.32601	15.2174
3	6.23947	10.36	1.8315	8.99209
4	2.76379	6.7527	0.333	5.13834
5	1.12949	4.30853	0.0333	2.82609
6	0.418459	2.68498	–	1.48741
7	0.137218	1.62998	–	0.743707

Table 23 Comparison of the probability of different delay at m = 20, different n and different active slot setting strategies

Slots of delay	n = 10	n = 10, adjusted	n = 15	n = 15, adjusted
0	40.1263	50	53.6709	75
1	25.0058	26.3158	25.74	19.7368
2	15.1804	13.1579	11.8537	4.38596
3	8.95002	6.19195	5.21698	0.773994
4	5.10607	2.70898	2.18209	0.0967492
5	2.8066	1.08359	0.86159	0.00644995
6	1.47848	0.386997	0.318549	–
7	0.741613	0.119076	0.109188	–

algorithm 2 to establish a route based on the wireless sensor network of the first step which is asynchronously adjusted by algorithm 1.

Thirdly, apply algorithm 3 to continuously adjust the active slots of the nodes in the asynchronously adjusted wireless sensor network obtained in the previous second step. After the adjustment, the active slot of each sensor is shown in Table 31.

Fourthly, we simulate the sensing and routing procedure by using the algorithms proposed above. At first, fire points will be randomly placed on the network shown in Fig. 18 and Table 31. And fire events may occur in any slot during the cycle. To do so, eighteen fire points were placed on the initial network and the wireless sensor networks by our Bi-adjusting methods, as shown in Fig. 19 and Fig. 20, which respectively according to Fig. 18 and Table 31. Then, simulate the process of random fires through computer programming. For the nodes on the network, the average EDL ($\overline{\mathrm{S}}^{i}$), the variance of EDL ($\sigma_{Si}^{2}$), the average DRD ($\overline{\mathrm{T}}^{i}$), the variance of DRD ($\sigma_{Ti}^{2}$) on the initial network, the average EDL($\overline{\mathrm{S}}^{b}$), the variance of EDL ($\sigma_{Sb}^{2}$), the average DRD ($\overline{\mathrm{T}}^{b}$), and the variance of DRD (σ_{Sb}^{2}), on the bi-adjusting network, are calculated by program simulation. Use p to represent the fire point.

The results of simulation experiments are as shown in Tables 32, 33, 34, 35, 36, 37, 38, 39, 40, and 41.

From Table 32, we can see that for the 18 fire points, after Bi-adjusting, the average EDL of six fire points decreases, the average EDL of seven fire points increases, and the EDL delay of five fire points is changeless. However, for the entire network, after bi-adjusting, the average EDL is reduced by 3.91%, which shows that bi-adjusting is effective for reducing the EDL. Variance represents the degree of deviation between the average EDL and EDL when the fire slots changes. According to the experiment, for different fire slots, the nodes that perceive events are different, and the more dispersed the active slot of the nodes that can perceive events, the larger the variance will be. From Table 32, the sum of the variance of EDL increases after the bi-adjusting, which indicates that the asynchronous adjustment works.

From Table 33, we can see that the average DRD of all fire points has decreased. For the entire network, after bi-adjusting, the average DRD is reduced by 56.22%. This shows that bi-adjusting is very effective for reducing DRD.

From Table 34, we can see that the average total delay of all fire points has decreased. For the entire network, the total average total delay decreased from 525.6 to 256.8,

Table 24 Average reduced delay slots of EDL after asynchronous adjustment

m	1	2	3	4	5	6
n = 5	0	0.03125	0.135802	0.269531	0.416	0.402392
n = 10	0	0	0.0173585	0.057291	0.113526	0.179841
n = 15	0	0	0.0022837	0.013394	0.0356556	0.0672197
n = 20	0	0	0	0.0031722	0.0115658	0.0263857
7	8	9	10	11	12	13
0.392614	0.385254	0.379515	0.374917	0.37115	0.368007	0.365347
0.252555	0.329517	0.409419	0.491434	0.484103	0.477961	0.472741
0.105694	0.149196	0.196384	0.246325	0.29836	0.35202	0.40696
0.04703	0.0724636	0.101703	0.133941	0.168553	0.205058	0.243089
14	15	16	17	18	19	20
0.363065	0.361086	0.359355	0.357826	0.356467	0.355251	0.354156
0.468252	0.464351	0.460931	0.457908	0.455216	0.452805	0.450633
0.462926	0.519725	0.514711	0.510266	0.506301	0.502741	0.49953
0.282364	0.322661	0.36381	0.405672	0.448139	0.491121	0.534546
21	22	23	24	25	26	27
0.353166	0.352265	0.351442	0.350688	0.349995	0.349354	0.348761
0.448666	0.446876	0.445241	0.443741	0.44236	0.441085	0.439904
0.496617	0.493964	0.491538	0.489311	0.487259	0.485362	0.483604
0.530735	0.527258	0.524073	0.521146	0.518446	0.515948	0.513631
28	29	30	31	32	33	34
0.34821	0.347698	0.347219	0.346771	0.346352	0.345957	0.345586
0.438807	0.437786	0.436832	0.435939	0.435102	0.434316	0.433575
0.481971	0.480449	0.479027	0.477696	0.476447	0.475274	0.474168
0.511476	0.509466	0.507587	0.505828	0.504176	0.502623	0.50116
35	36	37	38	39	40	41
0.345236	0.344906	0.344593	0.344297	0.344016	0.343749	0.343495
0.432877	0.432218	0.431594	0.431003	0.430442	0.429909	0.429402
0.473126	0.472141	0.471209	0.470326	0.469487	0.468691	0.467933
0.49978	0.498475	0.497239	0.496068	0.494957	0.4939	0.492894
42	43	44	45	46	47	48
0.343253	0.343022	0.342802	0.342592	0.34239	0.342198	0.342013
0.428919	0.428458	0.428019	0.427598	0.427197	0.426812	0.426443
0.467211	0.466522	0.465865	0.465236	0.464635	0.464059	0.463507
0.491936	0.491022	0.490149	0.489314	0.488516	0.487751	0.487018
49	50					
0.341836	0.341666					
0.426089	0.42575					
0.462978	0.46247					
0.486314	0.485638					

reduced by 51.14%. This shows that the bi-adjusting is effective for reducing the total delay.

In order to prove that the bi-adjusting is better than the asynchronous adjustment method and the continuous adjustment method in reducing EDL and DRD, only make

Table 25 The percentage of EDL slots reduced by asynchronous adjustment to EDL with slots randomly arranged, when the number of nodes and slots are different

m	1	2	3	4	5	6	7
n = 5	100	100	100	100	100	70.7119	54.083
n = 10	100	100	100	100	100	100	100
n = 15	100	100	100	100	100	100	100
n = 20	100	100	100	100	100	100	100
8	9	10	11	12	13	14	15
43.519	36.2762	31.0297	27.0685	23.9795	21.5077	19.4875	17.8072
100	100	100	84.1901	72.4426	63.4153	56.2878	50.5336
100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100
16	17	17	19	20	21	22	23
16.3888	15.1761	14.128	13.2133	12.4084	11.6949	11.058	10.4863
45.8006	41.8459	38.4964	35.6263	33.1415	30.971	29.0597	27.3648
89.1721	80.3232	72.9749	66.7881	61.5162	56.9765	53.0309	49.5733
100	100	100	100	100	91.7665	84.7006	78.5799
24	25	26	27	28	29	30	31
9.97023	9.50213	9.07566	8.68553	8.32732	7.99728	7.69224	7.40948
25.852	24.494	23.2685	22.1574	21.1456	20.2205	19.3716	18.5899
46.5208	43.808	41.3827	39.2026	37.233	35.4457	33.817	32.3271
73.2335	68.5284	64.3598	60.6438	57.3128	54.3119	51.5957	49.1268
32	33	34	35	36	37	38	39
7.14665	6.90172	6.67294	6.45876	6.25784	6.06899	5.89116	5.7234
17.8679	17.1991	16.5778	15.9993	15.4592	14.954	14.4804	14.0355
30.9593	29.6995	28.5357	27.4574	26.4558	25.5231	24.6524	23.8381
46.8737	44.8102	42.9139	41.1658	39.5496	38.0512	36.6586	35.3611
40	41	42	43	44	45	46	47
5.5649	5.41491	5.27277	5.13787	5.00967	4.8877	4.77151	4.6607
13.6168	13.2222	12.8495	12.4971	12.1633	11.8467	11.5461	11.2602
23.0747	22.3578	21.6832	21.0475	20.4473	19.8799	19.3426	18.8331
34.1496	33.016	31.9531	30.9547	30.0151	29.1295	28.2933	27.5027
48	49	50					
4.5549	4.45379	4.35706					
10.988	10.7286	10.481					
18.3494	17.8896	17.4519					
26.754	26.0442	25.3702					

asynchronous adjustment or connection adjustment to the active slot of nodes in the same wireless sensor networks.

From Table 35, we can see that for the 18 fire points, after bi-adjusting, the average EDL of five fire points decreases, the average EDL of six fire points increases, and the EDL delay of seven fire points is changeless. For the entire network, after bi-adjusting, the average EDL is increased by 0.82%.

This indicates that for the network as shown in Table 39, continuous adjustment of nodes on the basis of asynchronous adjustment will increase the average EDL.

Table 26 The average slots of DRD, when *n* = 5, 10, 15, 20, and *m* increased from 1 to 50

m	1	2	3	4	5	6	7	8
n = 5	4	6	8	10	12	14	16	18
n = 10	9	13.5	18	22.5	27	31.5	36	40.5
n = 15	14	21	28	35	42	49	56	63
n = 20	19	28.5	38	47.5	57	66.5	76	85.5
9	10	11	12	13	14	15	16	17
20	22	24	26	28	30	32	34	36
45	49.5	54	58.5	63	67.5	72	76.5	81
70	77	84	91	98	105	112	119	126
95	104.5	114	123.5	133	142.5	152	161.5	171
18	19	20	21	22	23	24	25	26
38	40	42	44	46	48	50	52	54
85.5	90	94.5	99	103.5	108	112.5	117	121.5
133	140	147	154	161	168	175	182	189
180.5	190	199.5	209	218.5	228	237.5	247	256.5
27	28	29	30	31	32	33	34	35
56	58	60	62	64	66	68	70	72
126	130.5	135	139.5	144	148.5	153	157.5	162
196	203	210	217	224	231	238	245	252
266	275.5	285	294.5	304	313.5	323	332.5	342
36	37	38	39	40	41	42	43	44
74	76	78	80	82	84	86	88	90
166.5	171	175.5	180	184.5	189	193.5	198	202.5
259	266	273	280	287	294	301	308	315
351.5	361	370.5	380	389.5	399	408.5	418	427.5
45	46	47	48	49	50			
92	94	96	98	100	102			
207	211.5	216	220.5	225	229.5			
322	329	336	343	350	357			
437	446.5	456	465.5	475	484.5			

From Table 36, we can see that the average DRD of all fire points has decreased. For the entire network, the average DRD decreased from 465.4 to 207.7, reduced by 55.37%. This shows that the bi-adjusting is effective for reducing the total delay. This indicates that the bi-adjusting is more beneficial to reduce DRD than only asynchronous adjustment.

From Table 37, we can see that the average total delay of all fire points has decreased. For the entire network, the total average total delay decreased from 514.1 to 256.8, reduced by 50.05%. This indicates that the bi-adjusting is better than only asynchronous adjustment in reducing delay.

From Table 38, we can see that for the 18 fire points, after bi-adjusting, the average EDL of five three points decreases, the average EDL of six fire points increases, and the EDL delay of nine fire points is changeless. For the entire network, after bi-adjusting, the average EDL is reduced by 2.77%. This indicates that the bi-adjusting is more beneficial to reduce EDL than only continuous adjustment.

Table 27 The average slots changes of DRD reduction, when n = 5,10,15,20, and m increased from 1 to 50

m	1	2	3	4	5	6	7	8
n = 5	0	2	4	6	8	10	12	14
n = 10	0	4.5	9	13.5	18	22.5	27	31.5
n = 15	0	7	14	21	28	35	42	49
n = 20	0	9.5	19	28.5	38	47.5	57	66.5
9	10	11	12	13	14	15	16	17
16	18	20	22	24	26	28	30	32
36	40.5	45	49.5	54	58.5	63	67.5	72
56	63	70	77	84	91	98	105	112
76	85.5	95	104.5	114	123.5	133	142.5	152
18	19	20	21	22	23	24	25	26
34	36	38	40	42	44	46	48	50
76.5	81	85.5	90	94.5	99	103.5	108	112.5
119	126	133	140	147	154	161	168	175
161.5	171	180.5	190	199.5	209	218.5	228	237.5
27	28	29	30	31	32	33	34	35
52	54	56	58	60	62	64	66	68
117	121.5	126	130.5	135	139.5	144	148.5	153
182	189	196	203	210	217	224	231	238
247	256.5	266	275.5	285	294.5	304	313.5	323
36	37	38	39	40	41	42	43	44
70	72	74	76	78	80	82	84	86
157.5	162	166.5	171	175.5	180	184.5	189	193.5
245	252	259	266	273	280	287	294	301
332.5	342	351.5	361	370.5	380	389.5	399	408.5
45	46	47	48	49	50			
88	90	92	94	96	98			
198	202.5	207	211.5	216	220.5			
308	315	322	329	336	343			
418	427.5	437	446.5	456	465.5			

From Table 39, we can see that for the 18 fire points, after bi-adjusting, the average DRD of five nine points decreases, the average DRD of seven fire points increases, and the DRD of two fire points is changeless. For the entire network, after bi-adjusting, the average DRD is reduced by 5.76%. This indicates that the bi-adjusting is more beneficial to reduce DRD than only continuous adjustment, but the effect is not obvious.

Table 28 The parameter settings of the WSNs used in the experiment

Parameters	Setting
The number of nodes(*n*)	90
The number of slots in one cycle(*m*)	10
The number of fire points	18
Slot of fire event	Any slot within a cycle
Time required to spread a hop	1 slot

Table 29 The initial active slot of the 90 nodes according to Fig. 4

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
1	2	6	8	7	4	5	9	4	5
v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}
7	0	8	3	2	4	6	4	9	3
v_{21}	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}	v_{29}	v_{30}
2	0	0	2	4	9	9	8	8	5
v_{31}	v_{32}	v_{33}	v_{34}	v_{35}	v_{36}	v_{37}	v_{38}	v_{39}	v_{40}
5	4	0	7	3	5	2	2	4	3
v_{41}	v_{42}	v_{43}	v_{44}	v_{45}	v_{46}	v_{47}	v_{48}	v_{49}	v_{50}
7	2	5	6	1	3	0	8	1	0
v_{51}	v_{52}	v_{53}	v_{54}	v_{55}	v_{56}	v_{57}	v_{58}	v_{59}	v_{60}
4	1	7	6	1	4	5	6	8	6
v_{61}	v_{62}	v_{63}	v_{64}	v_{65}	v_{66}	v_{67}	v_{68}	v_{69}	v_{70}
3	7	9	3	2	4	9	9	1	1
v_{71}	v_{72}	v_{73}	v_{74}	v_{75}	v_{76}	v_{77}	v_{78}	v_{79}	v_{80}
3	6	7	8	5	7	5	9	6	9
v_{81}	v_{82}	v_{83}	v_{84}	v_{85}	v_{86}	v_{87}	v_{88}	v_{89}	v_{90}
0	7	8	7	1	0	7	4	4	5

The wireless sensor network is initialized, and the experiment begins

From Table 40, we can see that for the 18 fire points, after bi-adjusting, the average total delay of ten points decreases, the average total delay of five fire points increases, and the total delay of three fire points is changeless. For the entire network, after bi-adjusting, the average total delay is reduced by 4.43%. This indicates that the bi-adjusting is better than only asynchronous adjustment in reducing delay.

According to the number and percentage of reductions in the above several tables and the distance between the location of the sink node and the fire point arranged in the network, we find that the further away from the sink node, the greater the total

Table 30 Asynchronously adjusted active slot(red, bold, and italic indicate the adjusted active slot)

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	$\boldsymbol{v_9}$	v_{10}
1	2	6	8	7	4	5	9	***6***	5
v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	$\boldsymbol{v_{16}}$	v_{17}	$\boldsymbol{v_{18}}$	v_{19}	v_{20}
7	0	8	3	2	***5***	6	***7***	9	3
v_{21}	$\boldsymbol{v_{22}}$	$\boldsymbol{v_{23}}$	v_{24}	v_{25}	$\boldsymbol{v_{26}}$	$\boldsymbol{v_{27}}$	$\boldsymbol{v_{28}}$	$\boldsymbol{v_{29}}$	v_{30}
2	***3***	***9***	2	4	***0***	***5***	***1***	***3***	5
v_{31}	v_{32}	v_{33}	v_{34}	v_{35}	v_{36}	$\boldsymbol{v_{37}}$	$\boldsymbol{v_{38}}$	v_{39}	v_{40}
5	4	0	7	3	5	***6***	***7***	4	3
v_{41}	v_{42}	v_{43}	v_{44}	v_{45}	v_{46}	v_{47}	v_{48}	v_{49}	v_{50}
7	2	5	6	1	3	0	8	1	0
v_{51}	v_{52}	v_{53}	v_{54}	v_{55}	v_{56}	v_{57}	v_{58}	v_{59}	v_{60}
4	1	7	6	1	4	5	6	8	6
v_{61}	$\boldsymbol{v_{62}}$	v_{63}	v_{64}	v_{65}	v_{66}	v_{67}	v_{68}	$\boldsymbol{v_{69}}$	$\boldsymbol{v_{70}}$
3	***5***	9	3	2	4	9	9	***4***	***9***
v_{71}	v_{72}	v_{73}	v_{74}	v_{75}	v_{76}	v_{77}	v_{78}	v_{79}	v_{80}
3	6	7	8	5	7	5	9	6	9
v_{81}	$\boldsymbol{v_{82}}$	v_{83}	$\boldsymbol{v_{84}}$	v_{85}	v_{86}	v_{87}	$\boldsymbol{v_{88}}$	$\boldsymbol{v_{89}}$	v_{90}
0	***1***	8	***4***	1	0	7	***1***	***3***	5

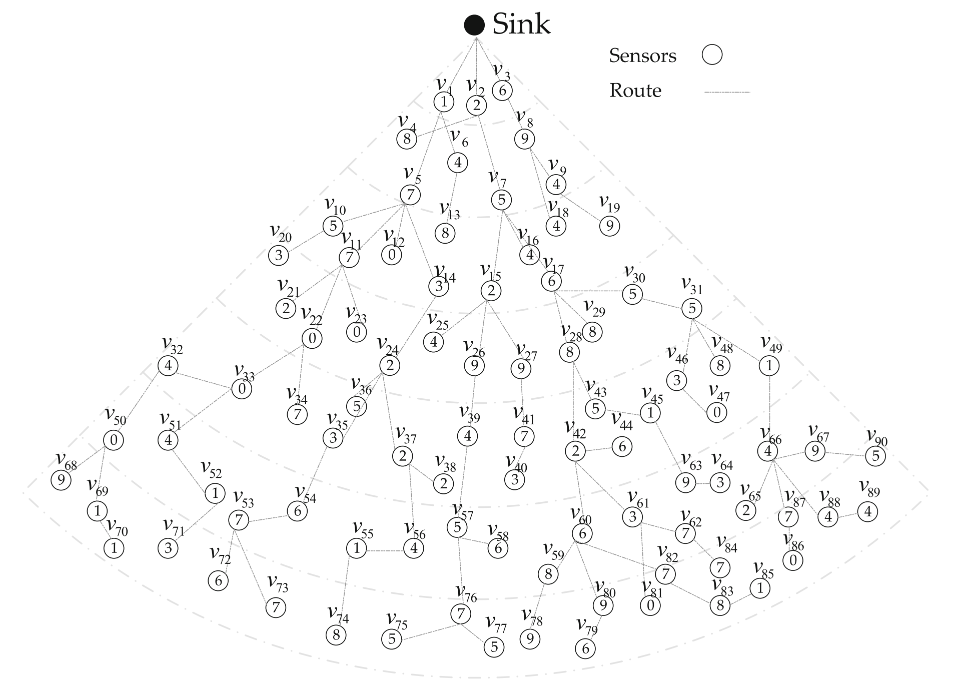

Fig. 18 Wireless sensor networks that establishes routing on initial setup

Table 31 Continuously adjusted active slot (red, bold, and italic indicate adjusted)

$\mathbf{v_{1}}$	$\mathbf{v_{2}}$	$\mathbf{v_{3}}$	v_{4}	$\mathbf{v_{5}}$	$\mathbf{v_{6}}$	$\mathbf{v_{7}}$	$\mathbf{v_{8}}$	$\mathbf{v_{9}}$	$\mathbf{v_{10}}$
0	***9***	***2***	8	***5***	***9***	***7***	***1***	***0***	***4***
$\mathbf{v_{11}}$	v_{12}	v_{13}	v_{14}	$\mathbf{v_{15}}$	v_{16}	$\mathbf{v_{17}}$	v_{18}	v_{19}	v_{20}
3	0	8	3	***6***	5	***4***	7	9	3
v_{21}	$\mathbf{v_{22}}$	v_{23}	v_{24}	v_{25}	$\mathbf{v_{26}}$	v_{27}	$\mathbf{v_{28}}$	v_{29}	$\mathbf{v_{30}}$
2	***8***	9	2	4	***8***	5	***8***	3	***3***
v_{31}	$\mathbf{v_{32}}$	$\mathbf{v_{33}}$	v_{34}	$\mathbf{v_{35}}$	v_{36}	$\mathbf{v_{37}}$	v_{38}	$\mathbf{v_{39}}$	v_{40}
2	***2***	***6***	7	***0***	5	***1***	7	***8***	3
v_{41}	$\mathbf{v_{42}}$	$\mathbf{v_{43}}$	v_{44}	$\mathbf{v_{45}}$	$\mathbf{v_{46}}$	v_{47}	v_{48}	$\mathbf{v_{49}}$	$\mathbf{v_{50}}$
7	***7***	***4***	6	***5***	***1***	0	8	***9***	***1***
$\mathbf{v_{51}}$	$\mathbf{v_{52}}$	$\mathbf{v_{53}}$	$\mathbf{v_{54}}$	$\mathbf{v_{55}}$	$\mathbf{v_{56}}$	$\mathbf{v_{57}}$	v_{58}	$\mathbf{v_{59}}$	$\mathbf{v_{60}}$
5	***4***	***8***	***9***	***9***	***0***	***8***	6	***0***	***4***
$\mathbf{v_{61}}$	v_{62}	$\mathbf{v_{63}}$	v_{64}	v_{65}	$\mathbf{v_{66}}$	$\mathbf{v_{67}}$	v_{68}	$\mathbf{v_{69}}$	v_{70}
6	5	***4***	3	2	***7***	***6***	9	***0***	9
v_{71}	v_{72}	v_{73}	v_{74}	v_{75}	$\mathbf{v_{76}}$	v_{77}	v_{78}	v_{79}	$\mathbf{v_{80}}$
3	6	7	8	5	***6***	5	9	6	***7***
v_{81}	$\mathbf{v_{82}}$	v_{83}	v_{84}	v_{85}	v_{86}	$\mathbf{v_{87}}$	$\mathbf{v_{88}}$	v_{89}	v_{90}
0	***3***	8	4	1	0	***1***	***4***	3	5

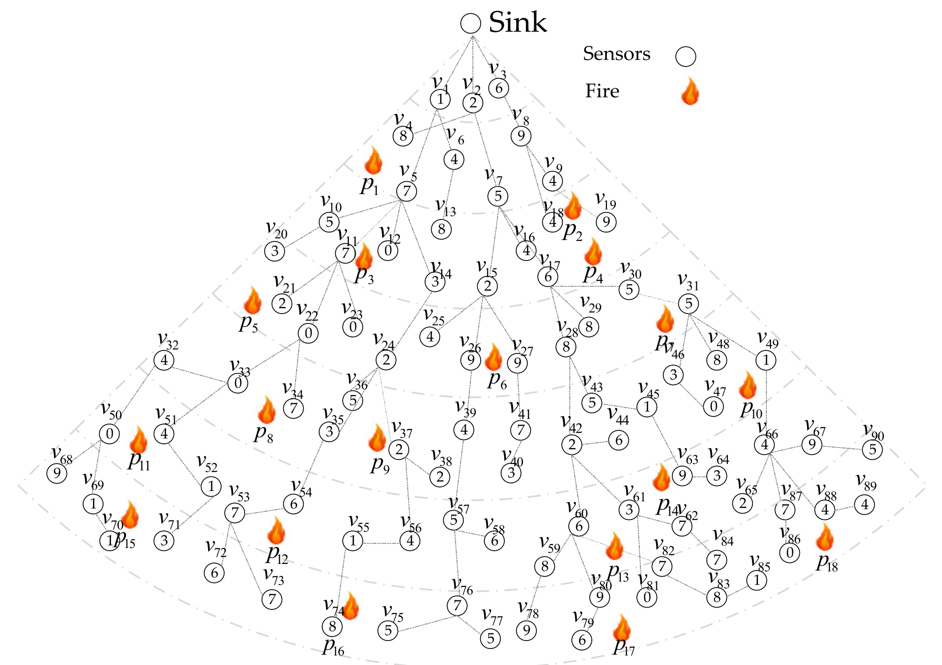

Fig. 19 Fire points placed on the WSNs after asynchronous adjustment (corresponds to Fig. 18)

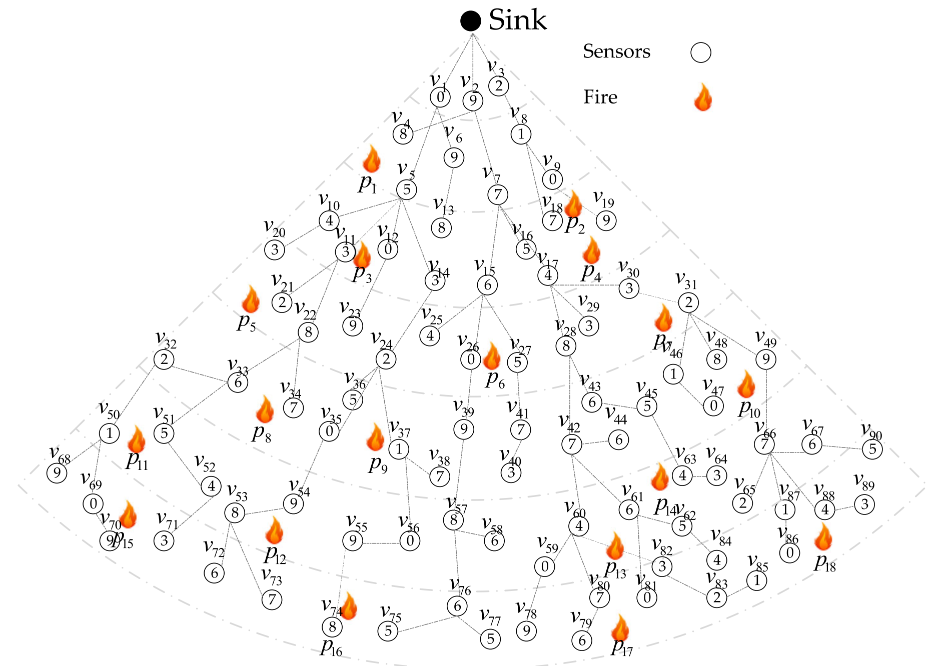

Fig. 20 Fire points placed on Bi-Adjusted WSNs after continuous adjustment (corresponds to Table 31)

Table 32 Average EDL, the variance of EDL, and percentage reduction for each fire point on the initial network and the bi-adjusting network

p	$\overline{S}^i$	σ^2_{Si}	$\overline{S}^b$	σ^2_{Sb}	Δ
p_1	3.6	8.27	2.4	4.27	33.33%
p_2	2	2.22	2.2	4.84	− 10.00%
p_3	2.2	3.07	2.4	4.27	− 9.09%
p_4	2.4	4.27	2	2.22	16.67%
p_5	4.5	9.17	4.5	9.17	0.00%
p_6	4.5	9.17	2.4	4.27	46.67%
p_7	4.5	9.17	3.6	8.27	20.00%
p_8	2.4	4.27	3.8	10.18	− 58.33%
p_9	2.2	4.84	1.6	2.04	27.27%
p_{10}	2.2	4.84	2.8	6.84	− 27.27%
p_{11}	2.1	2.77	2.1	2.77	0.00%
p_{12}	3.6	8.27	3.6	8.27	0.00%
p_{13}	2.4	4.27	2.9	6.32	− 20.83%
p_{14}	2.1	2.77	2.9	6.32	− 38.10%
p_{15}	2.9	6.32	1.8	3.07	37.93%
p_{16}	4.5	9.17	4.5	9.17	0.00%
p_{17}	1.8	3.07	1.8	3.07	0.00%
p_{18}	1.2	1.07	1.8	3.07	− 50.00%
Sum	51.1	96.97	49.1	98.41	3.91%

Table 33 Average DRD, the variance of DRD, and percentage reduction for each fire point on the initial network and the bi-adjusting network

p	$\overline{T}^i$	σ^2_{Ti}	$\overline{T}^b$	σ^2_{Tb}	Δ
p_1	4	0.00	3.8	3.73	5.00%
p_2	9.5	6.94	4.3	1.34	54.74%
p_3	10.5	2.50	9.1	2.10	13.33%
p_4	16.3	0.23	4	1.11	75.46%
p_5	19	0.00	8	0.00	57.89%
p_6	13	0.00	5.5	5.83	57.69%
p_7	17	0.00	6.9	0.10	59.41%
p_8	26.1	11.43	13.9	0.10	46.74%
p_9	18.3	1.57	11.9	7.21	34.97%
p_{10}	33.3	1.34	10.7	0.46	67.87%
p_{11}	39.4	4.27	17.4	4.27	55.84%
p_{12}	25.9	8.10	11.9	0.10	54.05%
p_{13}	31.1	11.43	14.6	0.71	53.05%
p_{14}	35.4	9.60	14.6	0.71	58.76%
p_{15}	49.6	0.71	19.7	3.57	60.28%
p_{16}	33	0.00	12	0.00	63.64%
p_{17}	44.7	2.90	21.7	3.57	51.45%
p_{18}	48.3	8.23	17.7	3.57	63.35%
Sum	474.4	/	207.7	/	56.22%

Table 34 Average total delay and percentage reduction for fire point on the initial network and the bi-adjusting network

p	$\overline{S}^i + \overline{T}^i$	$\overline{S}^b + \overline{T}^b$	Δ
p_1	7.6	6.2	18.42%
p_2	11.5	6.5	43.48%
p_3	12.8	11.5	10.16%
p_4	18.7	6	67.91%
p_5	23.5	12.5	46.81%
p_6	17.5	7.9	54.86%
p_7	21.5	10.5	51.16%
p_8	28.5	17.7	37.89%
p_9	20.5	13.5	34.15%
p_{10}	35.5	13.5	61.97%
p_{11}	41.5	19.5	53.01%
p_{12}	29.5	15.5	47.46%
p_{13}	33.5	17.5	47.76%
p_{14}	37.5	17.5	53.33%
p_{15}	52.5	21.5	59.05%
p_{16}	37.5	16.5	56.00%
p_{17}	46.5	23.5	49.46%
p_{18}	49.5	19.5	60.61%
Sum	525.6	256.8	51.14%

Table 35 Average EDL, the variance of EDL, and percentage reduction for each fire point on the WSNs after asynchronous adjustment and the bi-adjusting network

p	$\overline{S}^a$	σ^2_{Sa}	$\overline{S}^b$	σ^2_{Sb}	Δ
p_1	3.6	8.27	2.4	4.27	33.33%
p_2	2.2	4.84	2.2	4.84	0.00%
p_3	2.4	4.27	2.4	4.27	0.00%
p_4	2.1	2.77	2	2.22	4.76%
p_5	4.5	9.17	4.5	9.17	0.00%
p_6	2	2.22	2.4	4.27	− 20.00%
p_7	4.5	9.17	3.6	8.27	20.00%
p_8	2.4	4.27	3.8	10.18	− 58.33%
p_9	3.1	12.54	1.6	2.04	48.39%
p_{10}	2.2	4.84	2.8	6.84	− 27.27%
p_{11}	2.1	2.77	2.1	2.77	0.00%
p_{12}	3.6	8.27	3.6	8.27	0.00%
p_{13}	2	2.22	2.9	6.32	− 45.00%
p_{14}	2.1	2.77	2.9	6.32	− 38.10%
p_{15}	1.6	2.04	1.8	3.07	− 12.50%
p_{16}	4.5	9.17	4.5	9.17	0.00%
p_{17}	1.8	3.07	1.8	3.07	0.00%
p_{18}	2	3.11	1.8	3.07	10.00%
Sum	48.7	95.76	49.1	98.41	− 0.82%

Table 36 Average DRD, the variance of DRD, and percentage reduction for fire point on the WSNs after asynchronous adjustment and the bi-adjusting network

p	$\overline{T}^a$	σ^2_{Ta}	$\overline{T}^b$	σ^2_{Tb}	Δ
p_1	4	0.00	3.8	3.73	5.00%
p_2	11.3	9.12	4.3	1.34	61.95%
p_3	13.1	2.10	9.1	2.10	30.53%
p_4	17	0.00	4	1.11	76.47%
p_5	19	0.00	8	0.00	57.89%
p_6	14.5	6.94	5.5	5.83	62.07%
p_7	27	0.00	6.9	0.10	74.44%
p_8	23.1	2.10	13.9	0.10	39.83%
p_9	16.4	2.04	11.9	7.21	27.44%
p_{10}	33.3	1.34	10.7	0.46	67.87%
p_{11}	29.4	4.27	17.4	4.27	40.82%
p_{12}	29.1	47.21	11.9	0.10	59.11%
p_{13}	38.5	6.94	14.6	0.71	62.08%
p_{14}	35.4	9.60	14.6	0.71	58.76%
p_{15}	39.9	4.99	19.7	3.57	50.63%
p_{16}	23	0.00	12	0.00	47.83%
p_{17}	44.7	2.90	21.7	3.57	51.45%
p_{18}	46.7	14.90	17.7	3.57	62.10%
Sum	465.4	–	207.7	–	55.37%

Table 37 Average total delay and percentage reduction for each fire point on the WSNs after asynchronous adjustment and the bi-adjusting network

p	$\overline{S}^a+\overline{T}^a$	$\overline{S}^b+\overline{T}^b$	Δ
p_1	7.6	6.2	18.42%
p_2	13.5	6.5	51.85%
p_3	15.5	11.5	25.81%
p_4	19.1	6	68.59%
p_5	23.5	12.5	46.81%
p_6	16.5	7.9	52.12%
p_7	31.5	10.5	66.67%
p_8	25.5	17.7	30.59%
p_9	19.5	13.5	30.77%
p_{10}	35.5	13.5	61.97%
p_{11}	31.5	19.5	38.10%
p_{12}	32.7	15.5	52.60%
p_{13}	40.5	17.5	56.79%
p_{14}	37.5	17.5	53.33%
p_{15}	41.5	21.5	48.19%
p_{16}	27.5	16.5	40.00%
p_{17}	46.5	23.5	49.46%
p_{18}	48.7	19.5	59.96%
Sum	514.1	256.8	50.05%

Table 38 Average EDL, the variance of EDL, and percentage reduction for each fire point on the WSNs after continuous adjustment and the bi-adjusting network

p	$\overline{S}^c$	σ^2_{Sc}	$\overline{S}^b$	σ^2_{Sb}	Δ
p_1	2.4	4.27	2.4	4.27	0.00%
p_2	1.6	2.04	2.2	4.84	− 37.50%
p_3	2.4	4.27	2.4	4.27	0.00%
p_4	4.5	9.17	2	2.22	55.56%
p_5	4.5	9.17	4.5	9.17	0.00%
p_6	2.1	2.77	2.4	4.27	− 14.29%
p_7	3.6	8.27	3.6	8.27	0.00%
p_8	3.8	10.18	3.8	10.18	0.00%
p_9	1.4	1.82	1.6	2.04	− 14.29%
p_{10}	2.8	6.84	2.8	6.84	0.00%
p_{11}	2.9	6.32	2.1	2.77	27.59%
p_{12}	3.6	8.27	3.6	8.27	0.00%
p_{13}	2.4	4.27	2.9	6.32	− 20.83%
p_{14}	2.4	4.27	2.9	6.32	− 20.83%
p_{15}	2.2	4.84	1.8	3.07	18.18%
p_{16}	4.5	9.17	4.5	9.17	0.00%
p_{17}	1.8	3.07	1.8	3.07	0.00%
p_{18}	1.6	2.04	1.8	3.07	− 12.50%
Sum	50.5		49.1		2.77%

Table 39 Average DRD, the variance of DRD, and percentage reduction for fire point on the WSNs after continuous adjustment and the bi-adjusting network

p	$\overline{T}^c$	σ^2_{Tc}	$\overline{T}^b$	σ^2_{Tb}	Δ
p_1	4.4	0.93	3.8	3.73	13.64%
p_2	5.3	3.57	4.3	1.34	18.87%
p_3	9.1	2.10	9.1	2.10	0.00%
p_4	2	0.00	4	1.11	− 100.00%
p_5	8	0.00	8	0.00	0.00%
p_6	8.4	9.60	5.5	5.83	34.52%
p_7	8.9	0.10	6.9	0.10	22.47%
p_8	13.7	0.23	13.9	0.10	− 1.46%
p_9	18.1	4.54	11.9	7.21	34.25%
p_{10}	12.7	0.46	10.7	0.46	15.75%
p_{11}	16.6	0.71	17.4	4.27	− 4.82%
p_{12}	21.7	0.46	11.9	0.10	45.16%
p_{13}	9.1	2.10	14.6	0.71	− 60.44%
p_{14}	9.1	2.10	14.6	0.71	− 60.44%
p_{15}	18.7	0.46	19.7	3.57	− 5.35%
p_{16}	22	0.00	12	0.00	45.45%
p_{17}	13.7	3.57	21.7	3.57	-58.39%
p_{18}	18.9	6.32	17.7	3.57	6.35%
Sum	220.4	–	207.7	–	5.76%

Table 40 Average total delay and percentage reduction for each fire point on the WSNs after continuous adjustment and the bi-adjusting network

p	$\overline{S}^c + \overline{T}^c$	$\overline{S}^b + \overline{T}^b$	Δ
p_1	6.8	6.2	8.82%
p_2	6.9	6.5	5.80%
p_3	11.5	11.5	0.00%
p_4	6.5	6	7.69%
p_5	12.5	12.5	0.00%
p_6	10.5	7.9	24.76%
p_7	12.5	10.5	16.00%
p_8	17.5	17.7	− 1.14%
p_9	19.5	13.5	30.77%
p_{10}	15.5	13.5	12.90%
p_{11}	19.5	19.5	0.00%
p_{12}	25.3	15.5	38.74%
p_{13}	11.5	17.5	− 52.17%
p_{14}	11.5	17.5	− 52.17%
p_{15}	20.9	21.5	− 2.87%
p_{16}	24.3	16.5	32.10%
p_{17}	15.5	23.5	− 51.61%
p_{18}	20.5	19.5	4.88%
Sum	268.7	256.8	4.43%

delay reduced. In order to explore the relationship between the delay drop and the distance between the fire point and the sink node, the average hop $\overline{k}$ of the fire point is used to describe the distance between the fire point and the sink node:

$$\overline{k} = \frac{\sum\limits_{r<R_s} hop}{count} \tag{15}$$

That is, the average hop of a fire point is equal to the sum of the hop of the nodes which is centered on the fire point and in the range of the sensing radius divided by the number of nodes. Table 41 shows the average hop of 18 fires in Fig. 4.

In Table 41, we can see that the average hop of some fire points is the same.

In summarize, this section provides a complete description of the experiments the results, and analyses.

First, according to the setting of experimental parameters, the WSNs with 90 nodes are generated, and then active slots are randomly generated for 90 nodes. The asynchronous slot adjustment algorithm, routing algorithm, and continuous slot adjustment

Table 41 The average hop of 18 fire points

p	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
$\overline{k}$	2.00	3.33	3.00	3.33	4.00	4.00	4.50	5.00	5.00
p	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}
$\overline{k}$	6.33	6.50	6.50	6.00	6.50	8.33	8.50	7.33	8.33

algorithm proposed in this paper are all implemented in the experiment. The changes in the network after applying each algorithm are presented in figures and tables.

After applying the algorithm, a Bi-Adjusted network is obtained. According to the simulation of 18 fire points in the network, we calculated the delays in the two types of networks and compared the performance of the two types of networks. In order to prove that the bi-adjusting method is better than the other methods, we compare the performance of the bi-adjusting with the method of only making asynchronous adjustments and only making continuous adjustments.

7 Conclusion

In this paper, a bi-adjusting duty cycle schedule (BADCS) scheme that meets the green communication concept and can reduce the event detection latency and data routing delay of low-duty-cycle WSNs is proposed. According to the characteristics of the sensor node low duty cycle, our design method has made a breakthrough. Through theoretical analysis, the practical and useful of the method is proved. The experimental results show that the average event detection latency, average data routing delay, and average total delay are reduced by 3.91%, 56.22%, and 51.14% respectively. In most cases, the bi-adjusting results are better than the random arrangement of the active slot or the same as the random arrangement of the active slot, and in rare cases which are inferior to the random arrangement of the active slot. Compared to other related schemes on the same WSNs, BADCS is better than other related schemes.

Abbreviations
WSNs: Wireless sensor networks; IoT: Internet of Things; BADCS: Bi-adjusting duty cycle schedule; EDL: Event detection latency; DRD: Data routing delay; FNS: Forwarding node set; PPSS: Probability-based target prediction and sleep scheduling protocol; ASAM: Asynchronous slot adjustment request messages; RRM: Route Request Message; CRM: Creating a Route Message; DRM: Deleting a Routed Message; CSAM: Continuous time Slot Adjustment request Message

Acknowledgements
This work was supported in part by the National Natural Science Foundation of China (61772554 and 61472092).

Authors' contributions
Guopeng Li is the main author of the current paper. Fufang Li contributed to the conception and design of the study. Tian Wang, Jinsong Gui, and Shaobo Zhang commented the work. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]School of Computer Science and Engineering, Central South University, Changsha 410083, China. [2]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China. [3]Department of Computer Science and Technology, Huaqiao University, Xiamen 361021, China. [4]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China.

Received: 5 January 2020 Accepted: 14 July 2020
Published online: 31 August 2020

References
1. L. Chiaraviglio, F. Cuomo, M. Listanti, E. ManziaE, M. Santucci, Fatigue-aware management of cellular networks infrastructure with sleep modes. IEEE Transactions on Mobile Computing **16**(11), 3028–3041 (2017)
2. S. Sarkar, S. Chatterjee, S. Misra, Assessment of the suitability of fog computing in the context of Internet of Things. IEEE Transactions on Cloud Computing **6**(1), 46–59 (2018)

3. Li Z, Liu Y, Liu A, Wang S, & Liu H. Minimizing convergecast time and energy consumption in green internet of things. IEEE Transactions on Emerging Topics in Computing, DoI: https://doi.org/10.1109/TETC.2018.2844282, (IEEE Computer Society, 2018 in press).
4. X. Xiang, W. Liu, A. Liu, N. Xiong, Z. Zeng, Z. Cai, Adaptive duty cycle control–based opportunistic routing scheme to reduce delay in cyber physical systems. International Journal of Distributed Sensor Networks **15**(4), 1–21 (2019). https://doi.org/10.1177/1550147719841870
5. M. Huang, K. Zhang, Z. Zeng, *Wang T*, Liu Y. An AUV-assisted data gathering scheme based on clustering and matrix completion for smart ocean, IEEE Internet of Things Journal, 2020). https://doi.org/10.1109/JIOT.2020.2988035
6. Wang T, Cao Z, Wang S, Wang J, Qi I, Liu A, Xie M, Li X. Privacy-enhanced data collection based on deep learning for Internet of vehicles, IEEE Transactions on Industrial Informatics, 10.1109/TII.2019.2962844, (2019) (IEEE Computer Society, 2019 in press).
7. X. Liu, T. Qiu, B. Dai, L. Yang, A. Liu, *Wang J.* Swarm intelligence-based rendezvous selection via edge computing for mobile sensor networks, IEEE Internet of Things Journal, 2020). https://doi.org/10.1109/JIOT.2020.2966870
8. Y. Hu, M. Mong, K. Ota, A. Liu, M. Guo, Mobile target detection in wireless sensor networks with adjustable sensing frequency. IEEE Systems Journal **10**(3), 1160–1171 (2016)
9. Liu Q, Tian Y, Wu J, Peng T, Wang G. Enabling verifiable and dynamic ranked search over outsourced data. IEEE Transactions on Services Computing, DOI: 10.1109/TSC.2019.2922177, (IEEE, 2019 in press).
10. M. Wu, Y. Wu, C. Liu, Z. Cai, N. Xiong, Liu A, &Ma M. An effective delay reduction approach through portion of nodes with larger duty cycle for industrial WSNs. Sensors **18**(5), 1535 (2018). https://doi.org/10.3390/s18051535
11. W. Yang, W. Liu, Z. Zeng, A. Liu, G. Huang, N. Xiong, Z. Cai, Adding active slot joint larger broadcast radius for fast code dissemination in WSNs. Sensors **18**(11), 4055 (2018). https://doi.org/10.3390/s18114055
12. J. Tan, W. Liu, T. Wang, M. Zhao, A. Liu, S. Zhang, *A High-accurate content popularity prediction computational modelling for mobile edge computing by using matrix completion technology, transactions on emerging telecommunications technologies* (2019). https://doi.org/10.1002/ett.3871
13. T. Le Duc, D. Le, T, Zalyubovskiy V V, Kim D. Choo H. Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks. Pervasive and Mobile Computing **27**, 116–132 (2016)
14. H. Li, Y. Yang, T.H. Luan, X. Liang, L. Zhou, X.S. Shen, Enabling fine-grained multi-keyword search supporting classi-fied sub-dictionaries over encrypted cloud data. IEEE Trans-actions on Dependable and Secure Computing **13**(3), 312–325 (2015)
15. A. Baiocchi, L. Chiaraviglio, F. Cuomo, V. Salvatore, Joint management of energy consumption, maintenance costs, and user revenues in cellular networks with sleep modes. IEEE Transactions on Green Communications and Networking **1**(2), 167–181 (2017)
16. M. Tahir, R. Farrell, A cross-layer framework for optimal delay-margin, network lifetime and utility tradeoff in wireless visual sensor networks. Ad Hoc Networks **11**(2), 701–711 (2013)
17. Wang T, Peng Z, Wen S, Wang G. Wang B, Liu, A. A survey of fog computing in wireless sensor networks: concepts, applications and issues, Ad Hoc & Sensor Wireless Networks, 44, 109–130, (2019).
18. Y. Liu, M. Ma, X. Liu, N. Xiong, A. Liu, Y. Zhu, Design and analysis of probing route to defense sink-hole attacks for Internet of Things security. IEEE Transactions on Network Science and Engineering **7**(1), 356–372 (2020)
19. Q. Liu, P. Hou, G. Wang, T. Peng, S. Zhang, Intelligent route planning on large road networks with efficiency and privacy. Journal of Parallel and Distributed Computing **133**, 93–106 (2019). https://doi.org/10.1016/j.jpdc.2019.06.012
20. Liu Y, Zeng Z, Liu X, Zhu X, Bhuiyan M. A novel load balancing and low response delay framework for edge-cloud network based on SDN. IEEE Internet of Things Journal, DoI: https://doi.org/10.1109/JIOT.2019.2951857, (IEEE, 2019 in press).
21. X. Liu, A. Liu, Q. Deng, H. Liu, Large-scale programing code dissemination for software-defined wireless networks. The Computer Journal **60**(10), 1417–1442 (2017)
22. T. Li, M. Zhao, K. Won, Machine learning based code dissemination by selection of reliability mobile vehicles in 5G networks. Computer Communications. **152**, 109–118 (2020)
23. Li T, Liu W, Wang T, Zhao M, Li X, Ma M. Trust data collections via vehicles joint with unmanned aerial vehicles in the smart Internet of Things. Transactions on Emerging Telecommunications Technologies, DoI: https://doi.org/10.1002/ett.3956, (2020).
24. L. Cheng, J. Niu, C. Luo, L. Shu, L. Kong, Z. Zhao, Y. Gu, Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks. Computer Networks **134**, 66–77 (2018)
25. S. Yu, X. Liu, A. Liu, N. Xiong, Z. Cai, T. Wang, Adaption broadcast radius based code dissemination scheme for low energy wireless sensor networks. Sensors **18**(5), 1509 (2018). https://doi.org/10.3390/s18051509
26. A. Sultana, L. Zhao, X. Fernando, Efficient resource allocation in device-to-device communication using cognitive radio technology. IEEE Transactions on Vehicular Technology **66**(11), 10024–10034 (2017)
27. Y. Liu, A. Liu, N. Zhang, X. Liu, M. Ma, Y. Hu, Dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks. Journal of Network and Computer Applications **131**, 16–27 (2019)
28. Chen M, Wang T, Ota K, Dong M, Zhao M, Liu A. Intelligent resource allocation management for vehicles network: an A3C Learning Approach, Computer Communications. 151, 485–494, (2020).
29. A. Khatri, S. Kumar, O. Kaiwartya, N. Aslam, N. Meena, A.H. Abdullah, Towards green computing in wireless sensornetworks: Controlled mobility–aided balanced treeapproach. Int J Commun Syst **31**(7), e3463 (2018). https://doi.org/10.1002/dac.346318
30. K.S. Aanchal, O. Kaiwartya, et al., Green computing for wireless sensor networks: Optimization and Huffman coding approach. Peer-to-Peer Netw. Appl. **10**(3), 592–609 (2017). https://doi.org/10.1007/s12083-016-0511-y
31. H. Byun, J. Yu, Adaptive duty cycle control with queue management in wireless sensor networks. IEEE Transactions on Mobile Computing **12**(6), 1214–1224 (2013)
32. X. Xu, N. Zhang, H. Song, A. Liu, M. Zhao, Z. Zeng, Adaptive beaconing based MAC protocol for sensor based wearable system. IEEE Access **6**, 297700–229714 (2018)
33. K. Xie, X. Ning, X. Wang, D. Xie, J. Cao, G. Xie, J. Wen, Recover corrupted data in sensor networks: a matrix completion solution. IEEE Transactions on Mobile Computing **16**(5), 1434–1448 (2017)
34. Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, X. Lin, Healthdep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems. IEEE Transactions on Indus-trial Informatics **14**(9), 4101–4112 (2018)
35. Y. Zhao, T. Wang, S. Zhang, *Wang Y* (Towards mini-mum code dissemination delay through UAV joint vehicles for smart city, IET Communications, 2020). https://doi.org/10.1049/iet-com.2019.1205
36. A. Liu, Y. Hu, Z. Chen, An energy-efficient mobile target detection scheme with adjustable duty cycles in wireless sensor networks. International Journal of Ad Hoc and Ubiquitous Computing **22**(4), 203–225 (2016)

37. Y. Ren, Z. Zeng, T. Wang, S. Zhang, G. Zhi, A trust-based minimum cost and quality aware data collection scheme in P2P network. Peer-to-Peer Networking and Applications. (2020). https://doi.org/10.1007/s12083-020-00898-2
38. Peng M, Liu W, Wang T, Zeng Z. Relay selection joint consecutive packet routing scheme to improve performance for wake-up radio-enabled WSNs," Wireless Communications and Mobile Computing, 2020, Article ID 7230565, DoI:https://doi.org/10.1155/2020/7230565, (2020).
39. Y. Liu, A. Liu, Y. Hu, Z. Li, Y. Choi, H. Sekiya, J. Li, FFSC: An energy efficiency communications approach for delay minimizing in Internet of Things. IEEE Access **4**, 3775–3793 (2016)
40. D. Xu, W. Jiao, Z. Yin, J. Huang, Y. Peng, X. Chen, D. Fang, Z. Tang, Maximizing throughput for low duty-cycled sensor networks. Computer Networks **139**, 48–59 (2018)
41. L. Liang, X. Liu, Y. Wang, W. Feng, G. Yang, SW-MAC: A low-latency MAC protocol with adaptive sleeping for wireless sensor networks. Wireless Personal Communications **77**(2), 1191–1211 (2014)
42. Q. Liu, G. Wang, X. Liu, T. Peng, J. Wu, Achieving reliable and secure services in cloud computing environments. Computers & Electrical Engineering **59**, 153–164 (2017)
43. Wang X, Liu Z, Gao Y, Zheng X, Dang Z, Shen X. A near-optimal protocol for the grouping problem in RFID systems, IEEE Transactions on Mobile Computing (DOI:https://doi.org/10.1109/TMC.2019.2962125, (IEEE, 2019 in press).
44. Li F, Tang H, Zou Y, Huang Y, Feng Y, Peng L. Research on information security in text emotional steganography based on machine learning, Enterprise Information Systems, DOI: 10.1080/17517575.2020.1720827, (2020) (In press, Early access, Published online: 11 Feb 2020).
45. Q. Liu, Y. Guo, J. Wu, G. Wang, Effective query grouping strategy in clouds. Journal of Computer Science and Technology **32**(6), 1231–1249 (2017)
46. T. Wang, P. Wang, S. Cai, Y. Ma, A. Liu, M. Xie, A unified trustworthy environment based on edge computing in industrial IoT. IEEE Transactions on Industrial Informatics **16**(9), 6083–6091 (2020)
47. T. Wang, L. Qiu, A. Sangaiah, A. Liu, M. Bhuiyan, Y. Ma, Edge computing based trustworthy data collection model in the Internet of Things. IEEE Internet of Things Journal **7**(5), 4218–4227 (2020)
48. Q. Li, A. Liu, T. Wang, M. Xie, N. Xiong, Pipeline slot based fast rerouting scheme for delay optimization in duty cycle based M2M communications. Peer-to-Peer Networking and Applications **12**(6), 1673–1704 (2019)
49. W. Shi, W. Liu, T. Wang, Z. Zeng, G. Zhi, Adding duty cycle only in connected dominating sets for energy efficient and fast data collection. IEEE Access **7**(1), 120475–120499 (2019)
50. Wang T, Ke H, Zheng X, Wang K, Sangaiah A, Liu A. Big data cleaning based on mobile edge computing in industrial sensor-cloud, IEEE Transactions on Industrial Informatics. DoI: 10.1109/TII.2019.2938861, (IEEE, 2019 in press).
51. B. Jiang, G. Huang, T. Wang, J. Gui, J. Zhu, *Trust based energy efficient data collection with unmanned aerial vehicle in edge network, Transactions on Emerging Telecommunications Technologies* (2020). https://doi.org/10.1002/ett.3942
52. T. Wang, D. Zhao, S. Cai, A. Jia, A. Liu, Bidirectional prediction based underwater data collection protocol for end-edge-cloud orchestrated system. IEEE Transactions on Industrial Informatics **16**(7), 4791–4799 (2020)
53. Huang, M. ; Liu, W.; Wang, T.; Liu, A.; Zhang, S. A cloud-MEC collaborative task offloading scheme with service orchestration. IEEE Internet of Things Journal, doi:https://doi.org/10.1109/JIOT.2019.2952767, (IEEE, 2019 in press).
54. Y. Liu, X. Liu, A. Liu, N. Xiong, F. Liu, A trust computing based security routing scheme for cyber physical systems. ACM Transactions on Intelligent Systems and Technology **10**(6), 61 (2019). https://doi.org/10.1145/3321694
55. Li F, Li B, Huang Y, Feng Y, Peng L, Zhou N. Research on covert communication channel based on modulation of common compressed speech codec. Neural Computing and Applications, DOI:https://doi.org/10.1007/s00521-020-04882-y, (2020). (In press, Early access, Published online: 14 Apr 2020)
56. Huang M, Liu A, Xiong N, Wang T. Athanasios Vasilakos. An effective service-oriented networking management architecture for 5G-enabled Internet of Things, Computer networks, 173, 107208, (2020). https://doi.org/10.1016/j.comnet.2020.107208.
57. F. Wang, W. Liu, T. Wang, M. Zhao, M. Xie, H. Song, X. Li, A. Liu, To reduce delay, energy consumption and collision through optimization duty-cycle and size of forwarding node set in WSNs. IEEE Access **7**(1), 55983–56015 (2019)
58. X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, Z. Cai, Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks. Journal of Parallel and Distributed Computing **135**, 140–155 (2020)
59. Y. Chen, W. Liu, T. Wang, Q. Deng, A. Liu, H. Song, An adaptive retransmit mechanism for delay differentiated services in industrial WSNs. EURASIP Journal on Wireless Communications and Networking **2019**(1), 258 (2019). https://doi.org/10.1186/s13638-019-1566-2
60. X. Xiang, W. Liu, T. Wang, M. Xie, X. Li, H. Song, A. Liu, G. Zhang, Delay and Energy Efficient Data Collection Scheme based Matrix Filling Theory for Dynamic Traffic IoT. EURASIP Journal on Wireless Communications and Networking **2019**(1), 168 (2019). https://doi.org/10.1186/s13638-019-1490-5
61. Y. Liu, A. Liu, T. Wang, X. Liu, N. Xiong, An intelligent incentive mechanism for coverage of data collection in cognitive Internet of Things. Future Generation Computer Systems. **100**, 701–714 (2019)
62. X. Deng, Y. Jiang, Y. Yang, M. Lin, L. Yi, M. Wang, Data fusion based coverage optimization in heterogeneous sensor networks: a survey. Information Fusion **52**, 90–105 (2019)
63. Liu X, Wang T, Jia W, Liu A, Chi K. Quick convex hull-based rendezvous planning for delay-harsh mobile data gathering in disjoint sensor networks. IEEE Transactions on System, Man, and Cybernetics: Systems, DOI: https://doi.org/10.1109/TSMC.2019.2938790, (IEEE, 2019 in press).
64. B. Jiang, B. Ravindran, H. Cho, Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks. IEEE Transactions on Mobile Computing **12**(4), 735–747 (2013)
65. X. Liu, A. Liu, T. Qiu, B. Qiu, T. Wang, L. YANG, Restoring connectivity of damaged sensor networks for long-term survival in hostile environments. IEEE Internet of Things Journal **7**(2), 1205–1215 (2020)
66. Y. Hu, A. Liu, An efficient heuristic subtraction deployment strategy to guarantee quality of event detection for WSNs. The Computer Journal **58**(8), 1747–1762 (2015)
67. Teng H, Ota K, Liu A, Wang T, Zhang S. Vehicles joint UAVs to acquire and analyze data for topology discovery in large-scale IoT systems. Peer-to-Peer Networking and Applications. DoI: https://doi.org/10.1007/s12083-020-00879-5, (2020).

Publisher's Note