



Pipeline slot based fast rerouting scheme for delay optimization in duty cycle based M2M communications

Qiaoyan Li¹ · Anfeng Liu^{1,2} · Tian Wang³ · Mande Xie⁴ · Neal N. Xiong⁵

Received: 28 November 2018 / Accepted: 29 March 2019 / Published online: 25 June 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

In recent years, with the development of networked Cyber-Physical Systems (CPSs), wireless sensor networks (WSNs), as an important carrier of CPSs, has been applied more and more. In WSNs, many applications require low delay and high reliability for routing the sensing data to sink. Due to the lossy nature of wireless channels, rerouting schemes are often applied to ensure reliable data collection for mission-critical applications. However, rerouting together with multi-hop routing in duty cycle base WSNs will make designing a low delay routing scheme a challenge issue. In this paper, a Pipeline Slot based Fast Rerouting (PSFR) Scheme is proposed to reduce delay in duty cycle based WSNs. The main innovation points of PSFR scheme are as follows: (a) In duty cycle based WSNs, the major delay is caused by the sleep delay when nodes in the route forwarding to next hop node. Therefore, in PSFR scheme, we add a sequential active (SA) slot at the next hop node which is active at the next slot of the active slot of the previous node, which enables the previous node to forward packets to the next hop node in the slot right after receiving packet in active slot and greatly reduces sleep delay. (b) The second, in PSFR scheme, the backup path is designed beforehand in a less stringent pipeline active slot, so the packet can reach the sink with a relatively low delay when rerouting. (c) More importantly, in PSFR scheme, the added SA slots use the residual energy of peripheral nodes, which means they reduce the routing delay without decreasing network lifetime. After sufficient theoretical analysis and experiment, results show that the PSFR scheme can reduce the delay by more than 58.215% in the experiment networks without reducing the network lifetime, and the PSFR scheme can improve the energy utilization of network by more than 27.66%.

Keywords Wireless sensor networks · Pipeline slot · Rerouting · Delay · Lifetime

This article is part of the Topical Collection: *Special Issue on Networked Cyber-Physical Systems*

Guest Editors: Heng Zhang, Mohammed Chadli, Zhiguo Shi, Yanzheng Zhu, and Zhaojian Li

✉ Mande Xie
xiemd@zjgsuedu.cn

Qiaoyan Li
qiaoyanli@csu.edu.cn

Anfeng Liu
afengliu@mail.csu.edu.cn

Tian Wang
wangtian@hqu.edu.cn

Neal N. Xiong
dnxiong@126.com

¹ School of Computer Science and Engineering, Central South University, Changsha 410083, China

² Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, Jiangsu, China

³ Department of Computer Science and Technology, Huaqiao University, Xiamen 361021, Fujian Province, China

⁴ School of Computer Science and Information Engineering, Zhejiang Gongshang University, Zhejiang 310018, Hangzhou, China

⁵ Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

1 Introduction

With the rapid advancements in wireless technology, Wireless Sensor Networks (WSNs) are being widely deployed for mission-critical applications [1–4]. Mission-critical applications are classified as delay and loss intolerant [5–8]. The example of such applications include industrial site monitoring, detection of important dangerous places such as geological hazards, battlefield, life monitoring, fire monitoring and others [9–13]. Such types of mission-critical Wireless Sensor Network (WSN) applications require delay to be as low as possible and data reliability to be as high as possible, due to the fact that large delay or data loss may bring great threat to life and property, along with other disastrous consequences [1]. As wireless communication is essentially a kind of unreliable wireless link which is easy to fail, how to assure the low delay and high reliability of data routing in WSNs is a big challenge [14–18]. There are many challenge issues to be considered when designing a low delay and high reliability route [19, 20].

The first challenge issue is the unreliability of wireless link transmission, which often lead to failures in data routing. Routing failures can interrupt the data routing and increase delay. Therefore, some strategies are proposed to restore routing after data routing failures. Among them there is a effective method that rerouting to recovery the routing after the failure [1], since rerouting makes good use of the completed routing by selecting a backup routing from the failed node to the destination, avoiding the high energy consumption and long delay of restarting from the source node. However, designing a highly reliable rerouting scheme with low delay remains challenging.

The second challenge issue is the high routing delay, especially in duty cycle based WSNs. The nodes in WSNs are powered by batteries, so energy is quite valuable. In order to save energy and extend network lifetime as much as possible, applying duty cycle mode is an effective method [21, 22]. In duty cycle mode, nodes divide time into equal cycles T . Each cycle T is composed of k time slots. A node is awake only in one time slot normally, and sleeps in other time slots in a cycle [23, 24]. In this way, the duty cycle mode saves a lot of energy. But due to the problem that nodes can only send or receive data packets when it's awake thus unable to transmit data during the sleep, the sender has to be awake all through the waiting until the sleeping receiver wakes up to receive data. This delay results from the receiver node being in sleep state, so in this paper we call the delay caused by sleeping receiver which requires the sender to wait until the receiver turns to awake state to transmit data as sleep delay. Obviously, duty cycle mode saves node energy while increases delay [21–24].

Many research works have been done to reduce the delay in data transmission [25–27]. There are many factors that affect routing delay [28–31]. Data is routed to sink through multiple hops, so the delay from source to sink equals the sum of delays of each hop in the routing [32, 33]. There are many kinds of

data delay in each hop, first of all, the delay generated by data transmission [34, 35]. This kind of delay is related to data transmission rate of nodes and packet size. The delays caused by data transmission are related to the physical characteristics of the network, such as data transmission speed, data packet size [36, 37]. Generally speaking, this kind of delay is relatively small. The second kind of delay is queuing delay. Nodes transmit data at a limited speed, so some packets have to queue up and wait in the node's sending queue when packets arrive frequently, and this is how queuing delay is generated. Queuing delay is related to data load of nodes, the larger the load [38], the longer the queuing delay in network with sudden emergencies [21, 23]. The third kind is the delay related to data transmission protocol. The delay sizes of different protocols are very different from each other. For instance, lossy WSNs often adopt retransmission mechanism to ensure the reliability of data transmission, so a data packet may undergo multiple retransmissions, leading to large delay [39]. Some protocols adopt broadcasting routing or opportunistic routing [6]. Node broadcasts when sending data packets, and there are multiple receivers waiting to receive data at the same time. As long as one receiver get the packet, the transmission is successful [6]. As a result, data transmission in this way has a small delay. As for duty cycle based WSNs, sleep delay is the biggest delay [21–24] because sometimes the cycle T is in seconds, resulting in the delay in seconds, while data transmission is usually in milliseconds. Therefore, the key to reduce delay is to find a way to reduce sleep delay. The direct way is to extend node's duty cycle, such as select several instead of one time slot as active slot. In this way, when the sender sends data, the time waiting for receiver node will be reduced, thus reduce sleep delay effectively. But there is a problem that the number of active slots is almost linearly related to energy consumption, so in this method, reducing sleep delay multiply means increasing the energy consumption of nodes by several times, thus reducing the lifetime of nodes by several times. Although it decreases delay indeed, network lifetime is greatly cut down, limiting the widely use of this method.

Another way to reduce delay is to adjust the sequence of nodes' active slots [1, 8]. In a general duty cycle based WSN routing process, when a sender needs to transmit data to its next hop node, the delay is huge because the sender has to wait many slots until the receiver wakes up to transmit data if the sender's active slot is separated from the receiver's active slot by multiple slots. If we make the series of nodes on the routing path active in sequence, then this is what we call pipeline slot. In this method, nodes can directly transmit data to next hop node in the slot right after the one in which it receives data, which means data packets can be transmitted continuously without pauses in routing. Although this method reduces sleep delay a lot, there are some deficiencies still. This method adapts very well to fast routing of data due to the adjustment in nodes' active slots. However, changing node's active slot

may damage node's capacity to monitoring the environment, which is the fundamental function of a sensor node. For example, basically the nodes' active slots are required to be staggered in WSNs to sense the events in time, while in some applications the receiver node is asked to be awake at the same time in opportunistic routing to overcome the unreliability of wireless networks. This forced change of node active slot can reduce the delay of data routing, but at the cost of losing the main functions of sensor networks. So this method is also a strategy whose loss outweighs the gain.

Therefore, in this paper, a Pipeline Slot based Fast Rerouting (PSFR) Scheme is proposed to reduce delay in duty cycle based WSNs. The main innovation points of our work are as follows:

- (1) In Pipeline Slot based Fast Rerouting (PSFR) Scheme, the main method to reduce delay is to add a sequential active (SA) slot using the residual energy of nodes. The innovations make our scheme better than existing ones by adding SA slot are as follows. For one thing, we proved that pipeline slot in data routing is implementable by adding a SA slot to each node, without changing the existing active slot. For the other, SA slots are added using the residual energy of nodes, so they have no impact on network lifetime. Therefore, PSFR scheme can reduce delay greatly and avoid interfering with network lifetime or network sensing ability at the same time, which is very difficult to achieve in previous schemes.
- (2) A loose pipeline active slot based backup routing path planning method is proposed to reroute when the routing fails, thus further reduces data transmission delay and improves network performance. In the primary routing, the former and later nodes in the path make up a strict pipeline active slot. But when the primary routing failed, we still have a backup plan. Our scheme designed several backup paths beforehand for rerouting after failures. The primary routing can adopt the strict pipeline active slot way, while multiple backup routing paths can be taken for rerouting after one failure in the primary routing, making it difficult for backup routing to have a strict pipeline active slot pattern. In PSFR scheme, backup routing and primary routing are planned together, and the active slots of each pair of neighbor nodes in backup routing is as close as possible. Owing to the SA slot added in PSFR scheme, it is made possible to achieve a relatively loose pipeline active slot when designing backup routing. Thus, the data can be rerouted to sink at a fast speed when the routing fails.
- (3) More importantly, in PSFR scheme, the added SA slot uses the residual energy of nodes, reducing the routing delay without decreasing network lifetime. After sufficient theoretical analysis and experiment, results show that the PSFR scheme can reduce the delay by more than

58.215% in the experiment networks without reducing the network lifetime, and the PSFR scheme can improve the energy utilization of network by more than 27.66%.

The rest of this paper is organized as follows: Section 2 discusses existing researches on routing schemes in WSNs. In Section 3, the network model and problem statement are presented. Then, the PSFR scheme is introduced in Section 4. The parameter optimization and performance analysis of PSFR are presented in Section 5. The experimental results are given in Section 6. Finally, Section 7 provides conclusions.

2 Related work

With the rapid development of microprocessor technology [40–42], Internet of Things (IoT) [43–46] based on it and other related networks like edge network [47–49], cloud network [50, 51] got fast development, which bringing great changes to people's lifestyle [52]. Wireless sensor networks (WSNs) are often deployed in important application sites, such as battles, life monitoring and disaster monitoring [53, 54]. In such application scenarios, the data sensed by sensor nodes needs to be sent to sink quickly and reliably. Then the data sensed will be processed and analyzed, and a evaluation with further handling measures of possible dangers will be made to avoid the occurrence of disasters. There have been a lot of studies on schemes with low delay and high reliability for wireless sensor network. These researchs are discussed in the followings.

2.1 Research on delay optimization

The time from event occurrence to sensor node sensing data and the time in which data is transmitted to sink can all be counted as parts of total delay. The total delay includes many parts actually. To be more specific, the sensing delay and the delay of each hop in the multiple hop routing. The delay of each hop also contains two parts, which are queuing delay and transmission delay.

Sensing delay refers to the time from event occurrence to event being sensed by sensor node [24, 36]. Sensing delay is generated mainly for the reason that, in duty cycle based WSNs, due to the periodic sleep/awake state of nodes, the event won't be sensed until the node wakes up if it occurs in node's sleeping time. So sensing delay is closely related to the duty cycle of nodes, and the smaller the duty cycle (referring to the ratio of node's active slot time to the length of the whole cycle) is, the lower the sensing delay will be, otherwise higher. The network lifetime can be affected by extending duty cycle because it consumes more energy, so network designers often optimize network lifetime and sensing delay with tradeoff to

attain an overall performance optimization [24, 36]. Besides, in WSNs, nodes are usually deployed in a dense way, enabling multiple nodes to sense an event when it occurs, so for areas with a high density we can decrease the nodes' duty cycle to some degree [42].

The delay of data transmission is related to the protocol of data transmission. Because the link of wireless sensor network is unreliable, researchers propose a variety of protocols to guarantee the reliability of data transmission. The most important protocol of data transmission is send-wait ARQ protocol. In this protocol, after sender sends data to receiver, it waits for receiver to return ACK [39] that confirms the receipt of the packet. If sender does not receive the ACK returned by receiver within the expected time-out period, it considers that the data packet is lost in the transmission process, so it retransmits the data packet, which is called timeout retransmission. In the send-wait ARQ protocol, the above process has been going on all the time. When a data packet is successfully received by the receiver, or when the number of overtime retransmissions exceeds the predetermined resend threshold, it terminates. Because the time of one overtime retransmission is longer than that of one round trip transmission of the data packet, this kind of overtime retransmission mechanism produces a large delay [39].

The advantage of SW-ARQ protocol is that the protocol is simple and can be widely applied, but the delay is large. Especially in networks with bad wireless link quality, the average number of data retransmissions is large, leading to a large delay. Some studies have been done to improve the SW-ARQ protocol. In send-wait ARQ protocol, the transmission of a data packet occurs when the previous data packet is successfully sent or fails to transmit more than the maximum number of retransmissions. This sequential data transmission method still brings about a large delay. A later idea about improving this protocol is to change the data transmission to parallel transmission. That is, sending multiple data packets even when the previous packet did not respond an ACK. In this method, a window is set to record the unacknowledged packet information. The window size indicates the maximum number of unacknowledged packets that sender can allow. If the number of unacknowledged packets is less than the window size, the sender can continue to transmit packets to receiver. Once the receiver receives the ACK that confirms a packet is received, the sender clears the packet from the window. This method enables sender to transmit multiple packets at the same time, improves the network throughput and reduces delay. These protocols include GN protocol and other protocols [39].

It can be seen that the main cause of delay in data transmission is the instability of wireless links. Therefore, if the data transmission can be successful at one time, it can effectively reduce the delay. Therefore, reducing the delay is sometimes closely related to improving the reliability of data

transmission. Increasing transmission reliability is actually reducing delay, so these two methods are often adopted at the same time. Schemes applying these two methods include: Opportunity Routing Strategy [6]. In Opportunity Routing Strategy, because of the instability of wireless links, if the link fails when using sender-receiver mode, it needs to adopt a retransmit mechanism, which increases delay. However, if there are multiple receivers receiving data at the same time, we only need one receiver receives the data packet to transmit the data successfully. Therefore, this method can effectively reduce the number of retransmissions by raising receiving rate and thus reduces delay [6].

The delays discussed before are all under the condition that sensor nodes are awake all the time. In order to save energy, sensor nodes usually work in a duty cycle operation mode [21–24]. Therefore, when sender sends data, its receiver may be in a sleep state, so it needs to wait for its receiver to be awake before receiving data. Obviously, the longer a node is sleeping in a cycle, the larger the sleep delay. Therefore, the simplest way to reduce the sleep delay is to reduce the time of the node in the sleep state, that is, to increase the ratio of the time of the node in the awake state to the whole time cycle, or to increase the duty cycle. If the duty cycle increases to 1, which means the node is always awake, then the sleep delay is 0. However, although increasing the duty cycle of the node can reduce the sleep delay, it increases the energy consumption of the node and affects the network lifetime. Therefore, there is a tradeoff optimization relationship between the delay and the energy consumption.

Byun et al. [55] proposed an adaptive feedback mechanism to adjust duty cycle to ensure that delay is within the permissible range of applications. The main idea of their method is: when sink node receives the incoming data packet, if it finds that the end to end delay of the data packet is larger than the permissible threshold of the application, then it sends feedback to the path of the data packet source to increase duty cycle. The nodes on the packet source path increases the duty cycle when it receives the message from sink node. This reduces the delay of the packet to arrive at sink. Conversely, if the delay of the packet arriving at sink is less than the predetermined threshold, it means that the node can increase the duty cycle a bit more to save energy. Therefore, the corresponding reverse transmission of the message about reducing duty cycle to the nodes on the data source path increases the nodes' duty cycle on the data routing path, so as to minimize the delay [55]. Chen et al. [24] believe that if a feedback mechanism like [55] is adopted, the network lifetime will be decreased. The reason is as follows: In this mechanism, when the duty cycle of the nodes needs to be increased to reduce the delay, the nodes near sink tend to increase the duty cycle first, while those far from sink have less chance to increase the duty cycle. However, increasing duty cycle results in larger energy consumption while the nodes near sink node already have a

heavy data load, thus reduces the network lifetime. In the feedback mechanism, the magnitudes of increasing duty cycles of nodes are equal. However, because the nodes in the near sink region bear more data and consume more energy, the energy consumption of the network is unbalanced, which affects the network lifetime. Based on the above analysis, Chen et al. proposed an improved method: when the duty cycle needs to be increased to reduce the delay, the increase of the node's duty cycle is not equal. In their method, the nodes in the far sink region bear less data, so they have more energy to increase their cycle, while those in the near sink region increase less as they bear more data and have a larger energy consumption. In this way, by increasing the duty cycle of the nodes in the far sink region more and the ones in the near sink region less, the total delay in the network is lower or meets the application requirements, the energy consumption of the nodes in the near sink region is reduced, and the network lifetime is extended.

Delay is also related to the amount of data a node undertakes. Generally speaking, a node undertakes a large amount of data has a large queuing delay, so increasing its duty cycle improves the speed of data processing and reduces delays. Based on the above idea, Lee et al. [56] proposed an energy-efficient and delay-guaranteed algorithm. The solution is: the duty cycle of nodes with large data load is large, so the processing speed is fast, while the duty cycle of nodes with small data load is small, which can save energy.

Xie et al. [57] improved the above method. They found that according to Ref. [56], the nodes in the near sink region bear a large amount of data, thus setting up a larger duty cycle, while the nodes in the far sink region bear a small amount of data, so setting a smaller duty cycle. But this method is actually adverse to reducing delay. Because the nodes near sink bear more data and their duty cycles are larger, they consume more energy, while the nodes far from sink have less data and their duty cycles are smaller, so they consume less energy and have surpluses. So Rie et al. [57] proposed that the nodes in the far sink region should increase their duty cycles to make them larger than those of the nodes close to sink node. In this way, it reduces the delay of data transmission and makes full use of its energy, so this method has a good effect.

Naveen et al. [24] mainly discussed how to reduce delay in duty cycle based WSNs from the perspective of relay node selection. When sender sends data, those nodes that are closer to sink than the sender and within sender's transmitting radius can act as relay nodes, called forwarding node set. The nearest node to sink should be selected when the sender has data to send, because this enables sender to select the nearest node to sink in forwarding node set, and then move to sink in a hop route with the greatest distance, so that the data can arrive at sink with a small hop count and its delay is small. However, in duty cycle based WSNs, it is not easy to select the forwarding node nearest to sink for data forwarding. That's because the

node is sleep/awake periodically, and the forwarding node nearest to sink may also be in sleep state when sender has data to send. Therefore, the sender has two possible actions: one is waiting for the forwarding node to wake up, but this will increase delay due to the waiting; the other is that the sender does not wait but selects a node nearest to sink from the node in the awake state to forward. In this way, however, because the relay node selected is not the nearest node to sink, it may bring about more hops to sink, and its delay is not necessarily the smallest as well. Ref. [14] reduces it to a multi-objective optimization problem and gives an optimization algorithm.

2.2 Research on route reliability

Due to the unreliability of wireless links, data packets may fail due to node failure or security attacks in the process of routing. In this case, how to guarantee the successful arrival of routing at destination at a small cost is an important issue worth studying [58].

Rerouting mechanism is one of the earliest and effective methods used [1]. The idea of rerouting method is that when sender fails to route in the primary route, the route begins to find another route (called backup route) from the last valid node and rerouting to sink. The advantage of rerouting is obvious. It enables the cost of successful routing to be utilized, avoids the shortage of restarting routing from the source node, and effectively reduces the time required for routing, i.e. the delay.

Multipath routing mechanism is one of the most important routing methods to guarantee the reliability of routing as well [59]. Multipath routing mechanism is to route data packets from multiple paths to sink at the same time. Thus, as long as one of the routes can reach sink successfully, it can guarantee the success of routing and improve the reliability of routing. Another advantage of this method is that its delay is very small, which is even smaller than that of rerouting method. Besides, multipath routing strategy is easy to implement and can adopt general routing strategy, which can be applied to multiple networks. However, the biggest disadvantage of multi-path routing mechanism is that multiple routes are routed simultaneously, so the energy consumption of multi-path routing mechanism is many times of that of single-path routing mechanism, which seriously affects the lifetime of energy-intensive WSNs. Multi-path routing mechanism is also a strategy used to resist network attacks and improve network security. The key lies in initiating multiple routes at the same time, so attackers need to attack all routes at the same time to make the routing fail, which significantly increases the difficulty of attacking success and improves the security of the network.

In the multi-path routing method discussed above, complete data packets are transmitted in each route, which brings a great load to wireless sensor networks. Some researchers proposed using a (T, M) - threshold secret sharing mechanism for routing to ensure the success rate of routing. In such a mechanism, data packets are divided into M shares, which are routed to sink along

different routes. And sink only needs to receive T shares to recover a complete number of shares, while the attacker can't obtain the information of the data packet if the shares intercepted are less than T . Liu et al. [59] proposed the energy-efficient Disjoint Route (SEDR) scheme based on the above idea. SEDR strategy further improved the previous strategy. SEDR includes the following four stages: Data fragmentation, directional diffusion, "one-tier" random routing, and the shortest path. Among them, the "one-tier" random routing is the biggest innovation of SEDR strategy which is different from the previous strategies. The reason is as follows. In those strategies, when the packet is divided into M shares and then routed to sink along M routes, the M routes are as disjointed as possible and as far away from each other as possible with Random selection. In this way, the probability of attackers attacking M random distant routing paths at the same time is reduced, which can improve network security. However, in WSNs, due to the limited energy of nodes, it will consume more energy to route M paths as distant as possible. As a result, it is possible to reduce network lifetime. Therefore, in previous strategies, these M routes are generally not too far apart, so it is not reliable enough when encountering security attacks. The "one-tier" random routing proposed by SEDR can spread this M -path throughout the network, so it is very difficult for attackers to intercept M routes throughout the network, illustrating that the SEDR strategy can effectively improve the reliability of routing. The reason why SEDR does this is that it finds that there is a large amount of residual energy in the far sink region through analysis, so the shares can be spread to the whole network without affecting the lifetime.

Another way to ensure the reliability of routing is route regeneration method [60]. The idea of this method is: because wireless routing is unreliable and the success rate of routing declines with the increase of routing length, in order to ensure that data packets can be successfully transmitted to the destination, route regeneration method derives multiple routes to compensate for some failed routes after each routing period of the data packet [60]. Although in the routing process there are always failed routes disappear, successful routes will continue to generate new routes, so we can assure that there are always multiple routes in the process of data arriving at the destination, thus ensure the reliability of routing. The advantage of this method is that the delay of routing is relatively small. Compared with the method of generating M routes from the source node to ensure the reliability of routing, it costs less. This is because that sending M routes from the source node needs a larger M to ensure the reliability of routing. Successful routes arrive at the destination, while using route regeneration strategy, the number of routes regenerated each time is far less than M while achieving the same reliability, thus it has a smaller cost. However, the shortcoming of this method is that the control is more complex, and more routes are usually needed to be generated to ensure reliability than what are actually needed, which increases the cost of the system.

In duty cycle based WSNs, the rerouting method is very different from that in networks that suppose nodes to be awake all the time. In the rerouting method, when the primary route fails, the latest node which receives data needs to select a backup route for routing [1]. In previous pipeline slot methods, it is often assumed that data routing is reliable, so we only need to arrange the active slots of upstream and downstream nodes on the primary route as pipelines to enable the data to be routed to sink [8] with the smallest delay. However, when rerouting occurs, a big delay can still be generated because pipeline slots are not constructed between backup routes. Therefore, when using rerouting strategy, we should not only consider constituting a pipeline of active slot of the primary route nodes, but also try to make active slot of the backup route nodes a pipeline. Moreover, when constructing a pipeline, we should consider the arrangement of active slot globally. It is not an easy task because the active slots of nodes on different routes are different, and in the case of only one active slot, it is very difficult to make the nodes in multiple routes form pipelines. Since it is not technically guaranteed that pipeslots can be formed between multiple routes, as far as we know, there is no routing strategy that can establish pipeslots in both primary and backup routes.

The essential reason for improving routing reliability is the unreliability of wireless links. Therefore, improving routing reliability can also start with improving the quality of communication links. Firstly, the communication reliability of links is determined by the communication distance between nodes and sender's transmission power [58]. The success rate of data transmission between nodes decreases sharply as the distance between nodes increases, and the transmission power also decreases sharply. Some researchers have calculated the minimum transmission power that makes the energy consumption per bit of transmission the lowest, thus making the energy efficiency the highest. There are also some studies on the adjustment of transmission power and delay tradeoff optimization of nodes. Increasing the transmission power of nodes means that the transmission distance of data increases, so that the transmission distance of one hop increases, and the number of hops needed for data packets to reach sink decreases, thus reduces delay. Ju et al. [20] proposed energy conserving and transmission radius adaptive (ECTRA) scheme to reduce delay based on the above ideas. The main idea of ECTRA scheme is that when the network has plenty of solar energy, the nodes apply larger transmission power, so that the number of hops arriving at sink is less, and thus the delay is smaller.

2.3 Research on energy and lifetime

Energy and lifetime is one of the most critical performance indicators of WSNs [20–24]. Actually, it has a tradeoff relationship with other main indicators in the network. What is related to this paper is delay affecting routing. From the

previous discussion, we can see the following rule: if we want to reduce the delay of routing, the network will need to pay more energy and will reduce the network lifetime. For example, increasing duty cycle can reduce sleep and increase the rate of data forwarding at the same time [58], which can decrease the data transmission delay. In multi-path routing method, if the number of routing paths is more, the reliability of routing will be improved and the corresponding delays will be reduced. Improving the transmission power of nodes can, on the one hand, improve the receiving success rate of receiver nodes, thus reducing the number of data packets retransmitted and delay, on the other hand, transmit the packets farther with fewer hops of routing and reduce the delay.

From the above analysis, we can see that most of the current research either reduces the delay by increasing the energy consumption of the nodes, thus reducing the network lifetime, or accelerates the data routing by properly adjusting the active slots of the nodes on the route to form pipelines. In the latter way the active slots need to be re-established, but active slots are usually determined according to the requirements of applications and are not allowed to be adjusted, so they cannot be used in these applications. Therefore, this paper proposes a strategy that not only does not reduce network lifetime, but also can be used in various applications, which is of great significance.

3 System model and problem statement

3.1 The network model

In this paper, we consider a network model as follows: sensor nodes are evenly distributed in a circular region with radius R , including a sink node and n identical ordinary nodes $V = \{v_1, v_2, \dots, v_n\}$, where the sink node is located at the center of the circle. Therefore, the whole sensor network can be expressed as $W = \{sink, v_1, v_2, \dots, v_n\}$. Sensor networks can sense the surrounding environment. Each sensor node sends the sensing data to the next hop node, which has fewer hops from the sink node, and finally sends it to the sink node through multi-hop routing. Ordinary nodes are powered by the same batteries, with limited energy E_{node} , while the sink node's energy is infinite.

In order to save energy, the sensor network adopts the duty cycle mode, and nodes switch between sleep-wake modes. All nodes are divided by the same duty cycle, and a cycle is divided into $|T| = n$ time slots of the same duration. Therefore, a duty cycle can be expressed as slots $\{0, 1, 2, \dots, |T| - 1\}$. In one working cycle, the node wakes up in one of the slots, namely active slot, and sleeps in other slots to reduce energy consumption. Each node randomly chooses its own active slot. In this paper, a sequential active (SA) slot is added to the next slot of the last hop node's awake slot,

except the nodes with one hop from the sink node, thus reducing the delay waiting for the next hop node to wake up in data transmission.

In this model, there are three types of energy consumption of nodes. The first one is the energy E_{send} consumed by the node sending data packets to the next hop node; the second one is the energy $E_{receive}$ consumed by the node receiving data packets from the last hop node; and the third one is the energy E_{sleep} consumed by the node when it is in sleep state. Assume that all the data packets are of the same size. Nodes within a hop from the sink node spend more time on sending and receiving data in the active slot, so the energy consumption is fast and we don't add the sequential active slot to them. However, the energy consumption of peripheral nodes is slow. When the energy consumption of internal nodes leads to network death, the peripheral nodes usually still have much residual energy. Therefore, the energy is enough to add a sequential active (SA) slot to reduce delay. Details are given in Section 5.1.

Some of the symbols used in this paper are shown in Table 1.

3.2 The problem statements

Definition 1 Minimum transmission delay. In this paper, the transmission delay $\text{Delay}(v_i)$ of node v_i is defined as the number of slots between the last node sending out the data packet and the current node receiving the packet. Therefore, the total transmission delay can be expressed as $\sum_{v_i \in \text{route}} \text{Delay}(v_i)$. The aim of this paper is to reduce network transmission delay, which can be expressed as follows.

$$\min(D) = \min(\sum_{v_i \in \text{route}} \text{Delay}(v_i)) \quad (1)$$

Definition 2 Maximum energy utilization. In this paper, the energy utilization rate φ is defined as the ratio of the energy used by the network to the initial total energy of the network. We use e_i to denote the energy consumed by node v_i during the whole working process of the network, and e_m to denote the equal initial energy of each ordinary node. The aim of this paper is to maximize the energy utilization rate by adding SA slots using the residual energy of peripheral nodes. The maximum energy utilization rate can be expressed as follows.

$$\max(\varphi) = \max((\sum_{1 \leq i \leq n} e_i) / (\sum_{1 \leq i \leq n} e_m)) \quad (2)$$

Definition 3 Maximum network lifetime. In this paper, network lifetime L is defined as the time from network startup to the death of the first node. The energy consumption of a single node v_i includes the energy E_{send}^i consumed by sending data packets, the energy $E_{receive}^i$ consumed by receiving data

Table 1 Notations

Notation	Description
T	Working cycle of nodes
R_S	Message sending radius
S	The set of nodes whose distance to sink has been calculated
V	The set of all the nodes in the network
$d[v]$	Number of hops from node v to sink
d_m	The biggest number of hops from a node to sink in the network
M_d	The message to calculate $d[v]$
$A_S[v]$	The active slot of node v
$SA_S[v]$	The sequential active slot of node v
$flag[v]$	The flag to identify whether node v has a primary route to sink
M_p	The message to set up primary route
$SA_flag[v]$	The flag to identify whether node v has setted the SA slot
$end_flag[v]$	The flag to identify whether node v is a end node (a node that has no last hop node in any routes)
M_b	The message to set up backup route
M_{br}	The answer message of setting up backup route
E_m	The minimum energy above which the node can add branches (each node is limited to establishing at most two reverse primary route to other nodes)
E_{Dx}	The delay expectation of x hops
E_{DT}	The expectation of total transmission delay
H_N	Routing length (the number of hops)

packets and the energy E_{sleep}^i consumed by being in sleep state. The symbol e_m denotes the same initial energy for each ordinary node, thus the maximum network lifetime can be expressed as follows.

$$max(L) = \max_{0 \leq i \leq n} \left(e_m / \left(E_{send}^i + E_{receive}^i + E_{sleep}^i \right) \right) \tag{3}$$

Therefore, the problem to be addressed in this paper is to minimize the transmission delay D , maximize the network energy utilization rate φ , and maximize the network lifetime L . It can be summarized as follows.

$$\begin{cases} \min(D) = \min(\sum_{v_i \in route} Delay(v_i)) \\ \max(\varphi) = \max((\sum_{1 \leq i \leq n} e_i) / (\sum_{1 \leq i \leq n} e_m)) \\ \max(L) = \max_{0 \leq i \leq n} \left(e_m / \left(E_{send}^i + E_{receive}^i + E_{sleep}^i \right) \right) \end{cases} \tag{4}$$

4 The design of PSFR scheme

4.1 Research motivation

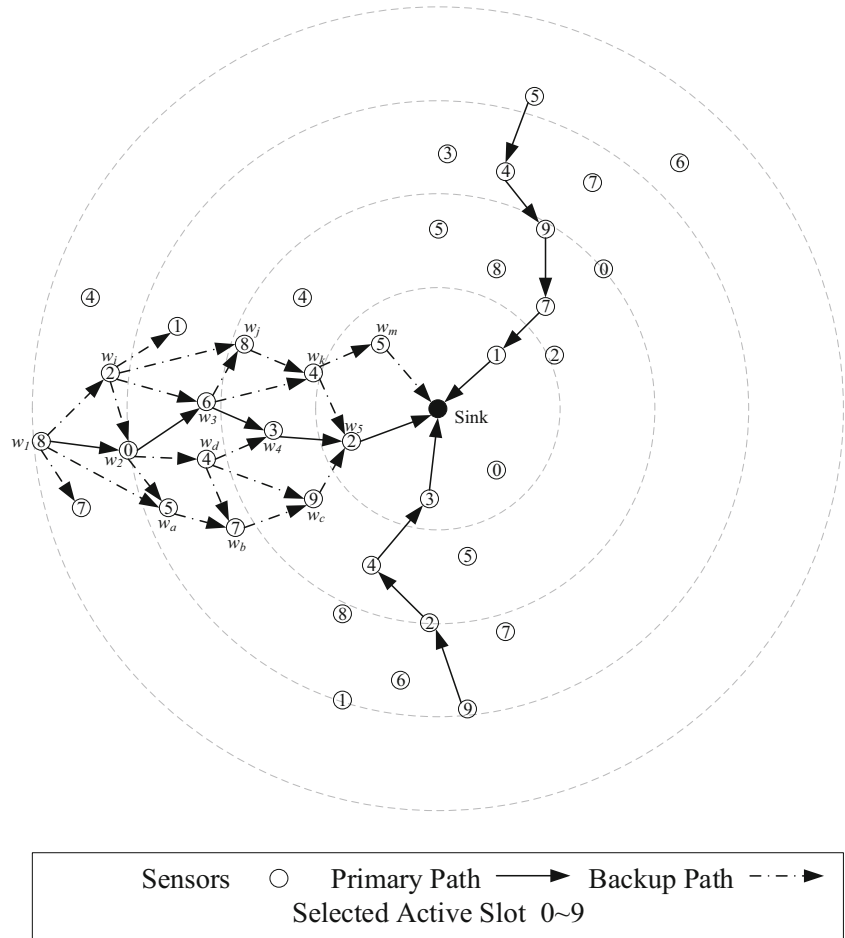
This section illustrates the motivation of this study by comparing different schemes in a specific data transmission process. In the traditional WSN rerouting scheme, each ordinary node wakes up only in the active slot selected randomly by itself within a working cycle. If it needs to transmit data, each node on the data transmission path needs to wait for the next

hop node to wake up to complete the data sending and receiving work. We call this traditional scheme random slot rerouting (RSR) scheme. On the one hand, networks using RSR scheme greatly saves the limited battery energy used by nodes. On the other hand, due to the discontinuity of active slots of the nodes in the data transmission path, the sender node needs to be awake for a long time until the next hop node wakes up. So there is a large sleep delay, which reduces the data transmission efficiency in RSR scheme.

Firstly, we consider the fault-free routing situation in the data transmission. A primary path and its backup path in a wireless sensor network using RSR scheme are shown in Fig. 1.

As shown in Fig. 1, if a duty cycle is divided into 10 slots and the number on each node is the random active slot selected by the node, the possible serial number of the active slot is 0–9. When node w_1 detects an event in the surrounding environment, it generates a packet and begins to transmit it to the central sink node. From node w_1 to sink, there is a pre-set primary path $w_1 \rightarrow w_2 \rightarrow w_3 \rightarrow w_4 \rightarrow w_5 \rightarrow sink$ and several backup paths. When w_2 wakes up in its active slot, which is the number 0 slot of the next working cycle, w_1 will connect to w_2 and transmit data to it. After w_2 receives the data, it waits until the next node w_3 wakes up in its active slot. The final transmission path is $w_1 \rightarrow w_2 \rightarrow w_3 \rightarrow w_4 \rightarrow w_5 \rightarrow sink$. In this data transmission process, each node has to wait until the next node wakes up at the active slot, so the transmission delay is very long, with a total of 20 slots needs to be delayed.

Fig. 1 A failure-free routing under RSR scheme



However, according to the Pipeline Slot based Fast Rerouting (PSFR) Scheme proposed in this paper, a sequential active (SA) slot is added utilizing the redundant energy of the peripheral nodes in WSNs, so that each node on the main path can wake up at the next slot of the awake slot of the previous hop node and ask if it needs to transmit data, which can greatly reduce the delay. The route after adding SA slot is shown in Fig. 2.

In Fig. 2, the numbers in red represent the SA slots of each node. At this point, if node w_1 detects an event and needs to transmit data to the central sink node, w_2 will add a SA slot in advance at the next slot of the active slot of w_1 , that is, the number 9 slot. Therefore, w_1 can establish a connection with w_2 to transmit data immediately. The next node w_3 also adds a SA slot at the next slot of the SA slot of w_2 in advance, which is the number 0 slot of the next working cycle. In this way, each node on the primary routing path adds a SA slot at the next slot of the SA slot of the previous node, thus the data can be transmitted to sink continuously, reducing the delay of each hop waiting for the next node to wake up. And this is the Pipeline Slot based Fast Rerouting (PSFR) Scheme in this article. With the PSFR scheme, the delay in slots of the whole data transmission process in Fig. 2 is 0.

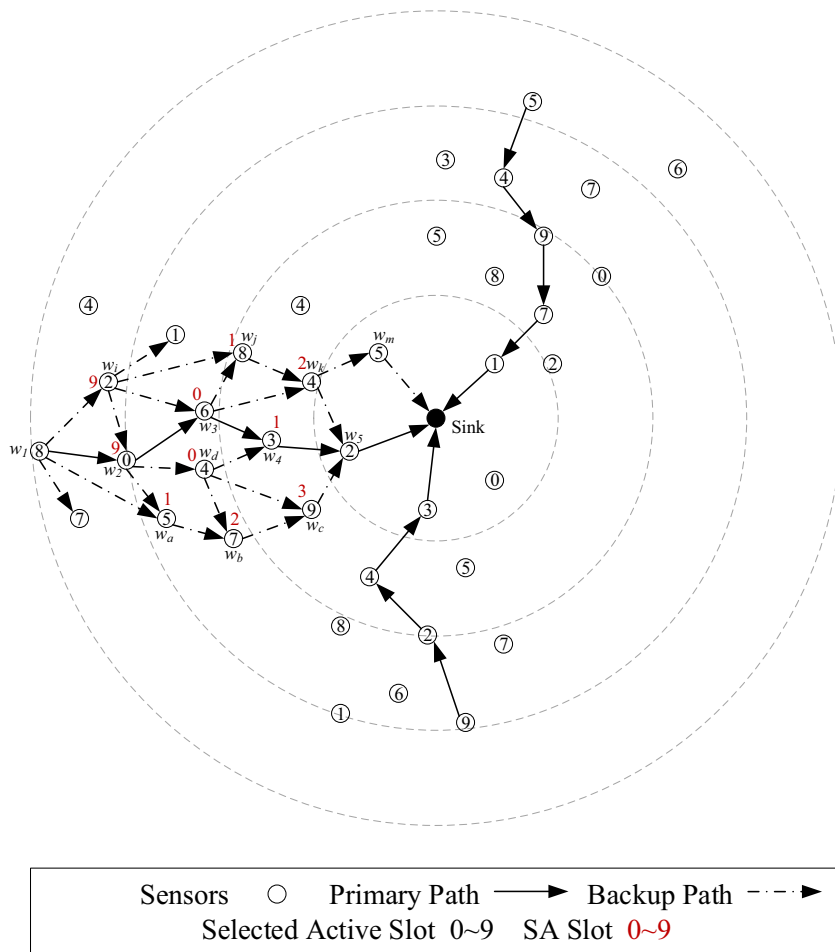
The comparison of the delay in the two strategies above can be expressed in the following Table 2.

In practical WSNs, data packets may fail due to the unreliability of wireless links in the routing process, so the considerations of the rerouting efficiency after failures are of great significance. In the following situation, we consider the case of rerouting after one failure in data transmission.

For wireless sensor networks using RSR method, the active slots of nodes on backup routes are as discontinuous as that of the nodes on primary routes. When a failure occurs on the primary path, the backup routing has a sleep delay similar to that of the primary routing, and the data transmission efficiency is still limited.

As is shown in Fig. 3, when RSR scheme is adopted, if the connection between w_1 and w_2 on the primary path in the sensor network fails, w_1 detects this situation and adopts the backup path in its routing table. When w_i wakes up in its active slot, the second slot in the next work cycle, w_1 will connect to w_i and transmit the data packet that needs to be sent to the sink node eventually. Node w_i waits until the next node wakes up in its active slot. The final transmission path of the data packet is $w_1 \rightarrow w_i \rightarrow w_3 \rightarrow w_4 \rightarrow w_5 \rightarrow sink$. In this rerouting transmission process, each node also has to wait

Fig. 2 A failure-free routing under PSFR scheme



until the next node wakes up at the active slot. The sleep delay of each hop is still very long, and a total delay of 20 slots is needed.

However, if the PSFR method is used in the same network, the rerouting delay after encountering failures will be greatly reduced, because the packets have more opportunities to be received earlier. Although the SA slot setting in backup routing in PSFR scheme is relatively loose, the efficiency of data transmission is still better than that of the traditional RSR scheme. The following Fig. 4 shows the rerouting of PSFR network after encountering the same failure.

It can be seen that when the connection between w_1 and w_2 fails, w_i on the backup path also has a pre-set SA slot in the next slot of w_1 's active slot, that is, the ninth slot in the same working cycle, so w_1 can immediately establish a connection with w_i without waiting. Then the next node w_3 also has a SA slot at the number 0 slot in the next working cycle. The data

can still be transmitted continuously without waiting for the next hop node to wake up, which reduces the delay in the whole data transmission process. In this data transmission, a total of 0 slots were delayed.

The delay comparison of the two strategies in the case of single failure can be expressed in Table 3.

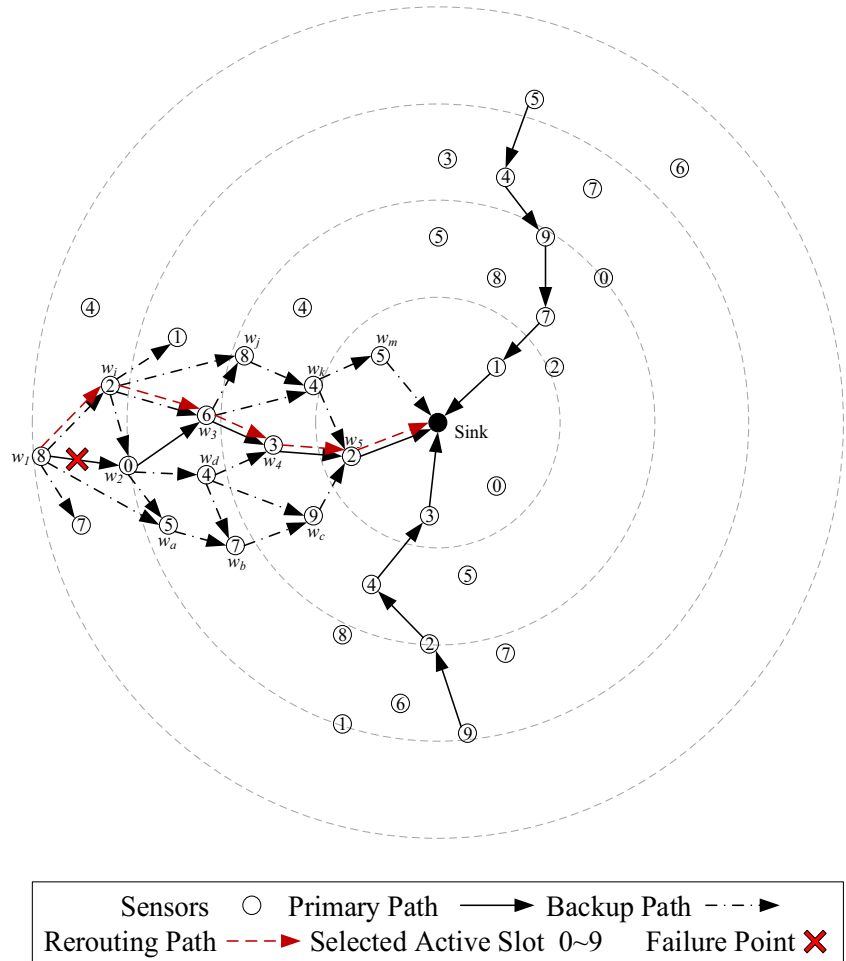
What if there are multiple failures in data transmission? For RSR method, the data transmission process of multiple rerouting is still discontinuous in time as in the case of failure-free. Figure 5 shows the case of multiple failures in routing using RSR scheme.

We can see from Fig. 5 that when a node wakes up only in a single slot in its working cycle, if there is another failure between node w_3 and w_4 and w_3 detects the situation, it needs to wait for w_j to wake up in the slot number 8 to establish a connection. Then w_j needs to wait for the next hop w_k to wake up. Therefore, the final transmission path of the data packet is $w_1 \rightarrow w_i \rightarrow w_3 \rightarrow w_j \rightarrow w_k \rightarrow w_m \rightarrow sink$. In this case, except for the hop from w_k to w_m that the active slot between the sender and the receiver happens to be continuous so that the sending node w_k does not have to wait for w_m to wake up, each of the other hops has a long waiting delay. The total slots of the delay waiting for data transmission is 12.

Table 2 Delay in a failure-free routing

Scheme	RSR	PSFR
Delay(slot)	20	0

Fig. 3 A single-failure rerouting under RSR scheme



When using the PSFR scheme, if there is another failure between w_3 and w_4 , w_3 uses backup path. The next hop w_j of backup path has a SA slot in the next slot of w_3 's active slot. Then w_k has a SA slot in the next slot, so the data packet can be transmitted continuously in the network with a small delay, and finally the transmission path is $w_1 \rightarrow w_i \rightarrow w_3 \rightarrow w_j \rightarrow w_k \rightarrow w_m \rightarrow sink$. Although the SA slot is not added at the nodes within one hop from sink, leading to the node that is two hops away from sink node needs to wait for the one-hop-away-from-sink node to wake up, in the overall data transmission process, the PSFR scheme still greatly reduces the delay of the sender node waiting in each hop, with a total delay of only two slots (Fig. 6).

The delay comparison of the two strategies in the case of multiple failures in routing can be expressed in Table 4.

In WSNs, the farther a node is away from sink, the smaller the data load the node undertakes. In a circular network, there are few nodes in the central area, and the nodes in the peripheral area far from the center are far more than those in the central area. However, the data detected by the peripheral nodes are all sent to the nodes in the central area to transmit to the sink node. Peripheral

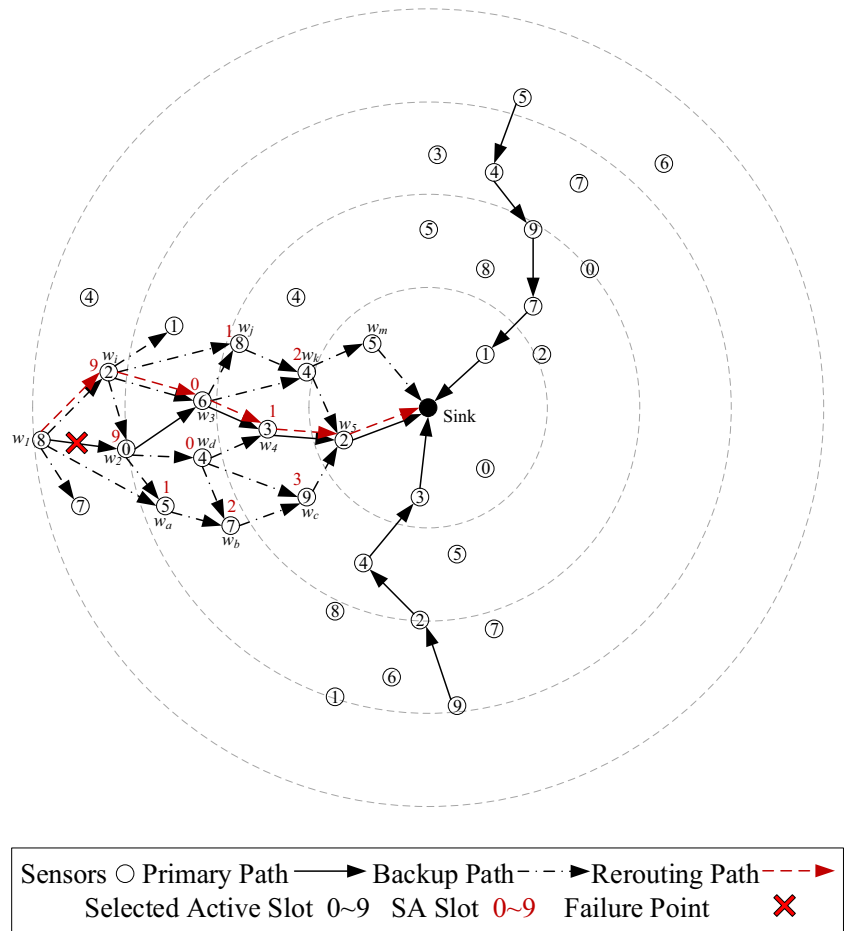
nodes only need to transmit data when they detect them or when the previous node needs transmission, and these data packets from all sides will eventually converge to a very limited number of ordinary nodes in the central area of the network. Therefore, these few nodes near sink undertake the transmission of all data packets in the network, which is much larger than the data load of the peripheral nodes. In WSNs, if the radius of the network is R , the transmitting radius of the node is r , the event rate is μ , the distance (in hops) between the node and sink is d , and s is the integer that makes $d + sr$ just less than R , the number of packets that each node bears can be expressed as follows [59].

$$N_d = ((s + 1) + sr(s + 1)/2d)\mu \tag{5}$$

According to Eq. (5), the relationship between the data load the node undertakes and the distance to the sink node is shown in Fig. 7 [59].

Obviously, when SA slot is not added, the energy consumption of peripheral nodes is much less than that of central nodes. When the energy of nodes near the center is exhausted,

Fig. 4 A single-failure rerouting under PSFR scheme



there is still a lot of energy left in peripheral nodes. But because the central node no longer works, the transmission efficiency of the network will decline rapidly and the network will die soon. Therefore, there is room for improvement in energy utilization in the traditional RSR scheme. So in the PSFR scheme, we use the residual energy of peripheral nodes to add a SA slot to reduce the delay waiting for the next node to wake up in the data transmission process.

4.2 Routing establishment and slot setting algorithms

4.2.1 SA slot

In WSNs, duty cycle mode is often used to save node energy. The duty cycle mode works as follows: the working time of nodes is divided into working cycles, and each working cycle T is divided into $|T| = n$ slots. The time slots in each work cycle

are marked from 0 to $|T| - 1$, that is, a work cycle can be expressed as a set of slots like $T = \{0, 1, 2, \dots, |T| - 1\}$. Each node randomly chooses a slot as the active slot. The node can establish a connection and transmit data in the active slot. In this paper, in addition to active slot, each peripheral node adds an additional SA slot, in which the node actively sends out a query signal. If there are nodes around that need to send messages, they can establish connections and transmit data like active slots; if there is no node response, the node switches to sleep state to save energy. A working cycle of a node with SA slot is shown in Fig. 8.

4.2.2 Establishment of primary route

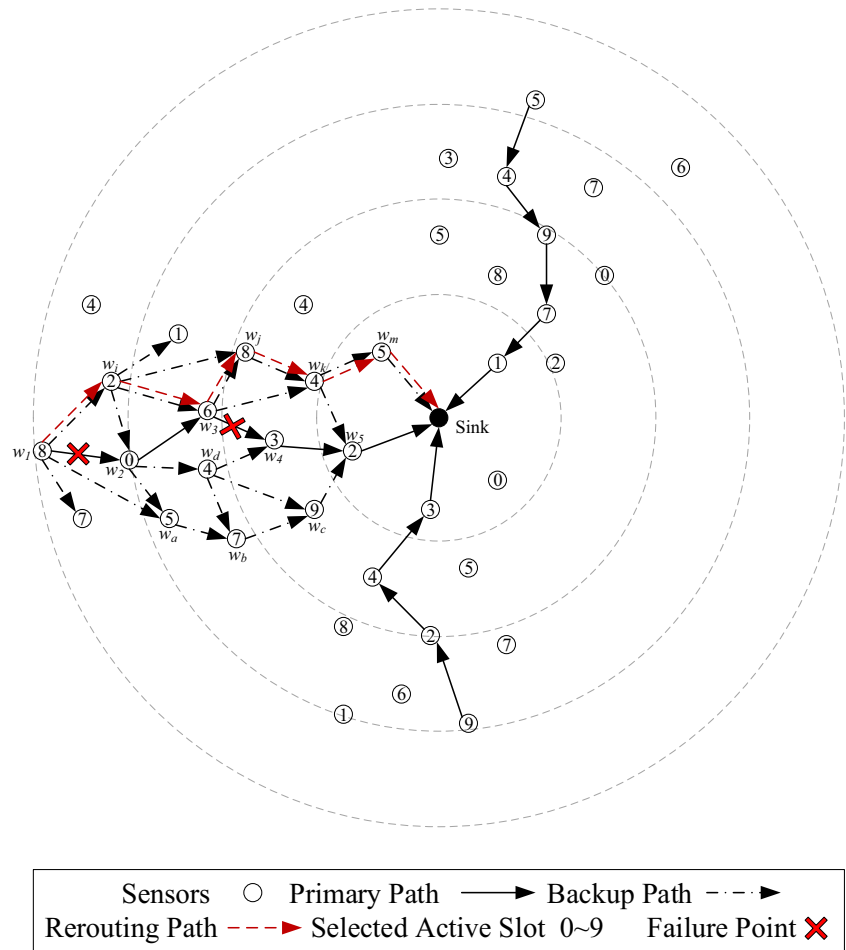
In the traditional RSR scheme, because the active slots of the nodes on the data transmission path are discontinuous, the node that obtains the data packet can only transmit the data when the next hop node wakes up.

One of the obvious shortcomings of RSR scheme is that the delay of data packet waiting for the next hop node to wake up is too long, and because the large amount of data that the central nodes near sink undertakes leads to fast energy consumption, the peripheral nodes often have a

Table 3 Delay in a single-failure rerouting

Scheme	RSR	PSFR
Delay(slot)	20	0

Fig. 5 A multi-failure rerouting under RSR scheme



large amount of energy surplus after the network death, and that is to say, the energy utilization rate of the network is not high.

To solve the above problems of RSR scheme, we propose a PSFR scheme, which adds a SA slot to the peripheral nodes with more than one hop away from the sink node. The SA slot of the outermost node is at the same slot as the active slot, then the SA slot of the next hop node is set to the next slot and so forth. In addition to waking up normally in active slot, the node also wakes up in SA slot and asks whether the surrounding nodes need to transmit data. If there is data to be transmitted, the node transmits it; if not, the node switches into sleep state again to save energy. In this way, if the peripheral node detects the data, it can continuously transmit the data packet to the sink node without or only with a little delay.

The routing establishment of PSFR method mainly includes three steps:

1. establish the primary routes
2. set the SA slot

3. Establish the backup routes

Before establishing the primary routes, we need to ascertain the number of hops of the nodes from sink. The sink node broadcasts the distance initialization message to the surrounding region of radius R_S , and the distance from the nodes received this message to sink is one hop. Then these nodes with one hop from sink broadcast distance initialization message with radius R_S as well. If some uninitialized nodes receive this message, the distance between them and sink is two hops, and so on. The specific process is shown in Algorithm 1.

After the initialization is completed, each node needs to establish a primary route to sink. In order to establish the primary route of each node to sink from inside to outside, we introduce a flag variable $flag[v]$ to mark whether the node v has a primary route to sink or not, in which situations the value of $flag[v]$ is 1 or 0 respectively. If the node detects a node that is closer to the sink node than itself and has established the primary route to the sink node, it can directly establish the primary route with this node, thus establishing the primary route to

reach the sink node. Each node establishes a primary route to a node that has less hops from sink and has a primary route to sink. The process of establishing reverse and forward primary routes is shown in Algorithms 2 and 3 (Fig. 9).

Algorithm 1: Initialize the number of hops from each node to sink

```

1: BEGIN
2:  $S = \emptyset$ 
3:  $d[\text{sink}] = 0$ 
4:  $d_m = 0$ 
5: FOR each node  $v \in V - \{\text{sink}\}$  DO
6:    $d[v] = \infty$ 
7: END FOR
8:  $S = S \cup \{\text{sink}\}$ 
9: WHILE  $V - S \neq \emptyset$  DO
10:  FOR each node  $u \in S$  &&  $d[u] = d_m$  DO
11:   Broadcast message  $M_d$  with radius  $R_S$ 
12:   FOR each  $v \in V - S$  DO
13:    IF  $v$  receives  $M_d$  DO
14:      $d[v] = d[u] + 1$ 
15:     IF  $d[v] > d_m$  DO
16:       $d_m = d[v]$ 
17:     END IF
18:     $S = S \cup \{v\}$ 
19:   END IF
20:  END FOR
21: END FOR
22: END WHILE
23: END

```

Algorithm 2: Establish reverse primary route

```

1: BEGIN
2:  $\text{flag}[\text{sink}] = 1$ 
3: FOR each  $u \in V - \{\text{sink}\}$  DO
4:    $\text{flag}[u] = 0$ 
5: END FOR
6: WHILE  $\exists u \in V, \text{flag}[u] = 0$  DO
7:    $u$  broadcast message  $M_p$ 
8:   FOR each  $v \in V - \{u\}$  DO
9:    IF  $v$  receives  $M_p$  DO
10:     IF  $v$  is the sink node DO
11:       $v$  establishes a reverse primary route to  $u$ 
12:      CALL PR
13:     ELSE IF  $\text{flag}[v] = 1$  &&
14:               $d[v] < d[u]$  &&  $E[v] > E_m$  DO
15:       $v$  establishes a reverse primary route to  $u$ 
16:     ELSE
17:      drop  $M_p$ 
18:     END IF
19:   END FOR
20: END WHILE
21: END

```

Algorithm 3: Establish forward primary route (PR)

```

1: BEGIN
2: FOR each  $v \in V$  DO
3:   FOR each  $u \in V - \{\text{sink}\}$  DO
4:     IF there is a reverse primary route from  $v$  to  $u$  DO
5:        $u$  establishes a forward primary route to  $v$ 
6:        $\text{flag}[u] = 1$ 
7:     END IF
8:   END FOR
9: END FOR

```

When a node in WSN detects an event, it passes the message to a node whose $d[v]$ value is smaller than it, and the data is passed to the node with smaller hop distance to sink in each hop, so that it can eventually be transmitted to sink.

4.2.3 Setting of SA slot

After the primary route from each node to sink is established, SA slot is set up. Apart from the fact that nodes with only one hop away from sink has no SA slot, each peripheral node has and only has one SA slot. We set a flag $\text{SA_flag}[v]$ to mark whether the node v has SA slot or not, where 1 means being set and 0 means not being set. Another flag, $\text{end_flag}[v]$, is used to record whether node v is an end node, that is, a node that has no last hop node in any routes, where 1 means it is an end node and 0 means it's not. For the most distant end nodes away from sink with radius over $3/4d_m$ hops, the SA slot is set at the same slot as active slot, and then the setting expands to the center of the network. The expanding process is discussed in the following two cases.

A. When the SA slot of the previous node has been set (Fig. 10)

In this case, SA slot is set as the next slot of SA slot of the previous hop node to form a continuous wake-up node chain, so that the last hop node does not need to wait for the current node to wake up when it has data to be transmitted.

B. When the SA slot of the next node has been set (Fig. 11)

In this case, the SA slot is set at the last slot of the SA slot of the next hop node so that the delay can be avoided when the data is transmitted to the next hop.

The algorithm is summarized in Algorithm 4.

Algorithm 4: Set SA slot

```

1: BEGIN
2: FOR each  $v \in V$  DO
3:   SA_flag[v] = 0
4:   end_flag[v] = 1
5:   FOR each  $u \in V - \{\text{sink}\}$  DO
6:     IF there is a forward primary route from u to v DO
7:       end_flag[v] = 0
8:     END IF
9:   END FOR
10:  IF end_flag[v] = 1 &&  $d[v] > 3/4dm$  DO
11:    SAs[v] = As[v]
12:    SA_flag[v] = 1
13:  END IF
14: END FOR
15: WHILE  $\exists w \in V, d[w] > 1$  && SA_flag[w] = 0 DO
16:  FOR each  $v \in V - \{\text{sink}\}$ 
17:    &&  $d[v] > 2$  && SA_flag[v] = 1 DO
18:    FOR each  $u \in V - \{\text{sink}\}$  DO
19:      IF there is a reverse primary route from u to v
20:        &&  $d[u] > 1$  && SA_flag[u] = 0 DO
21:        SAs[u] = (SAs[v]+1) mod |T|
22:      ELSE IF there is a reverse primary route from u to v
23:        &&  $d[u] > 1$  && SA_flag[u] = 1 DO
24:        IF (SAs[u] - (SAs[v] + 1)) mod 10
25:          > (SAs[v] + 1 - SAs[u]) mod 10 DO
26:          SAs[u] = (SAs[v] + 1) mod |T|
27:        ELSE IF there is a forward primary route from u to v
28:          && SA_flag[u] = 0 DO
29:          SAs[u] = (SAs[v] - 1) mod |T|
30:        END IF
31:      END FOR
32:    END FOR
33:  END WHILE
34: END

```

The network after setting SA slot is shown in Fig. 12.

4.2.4 Establishment and selection of backup route

When setting up backup route, each node sends a backup routing request message M_b to the nodes with the same or smaller distance to sink and not on its own primary route to sink. After receiving the request, the eligible nodes in the surrounding signal range establishes a backup route and sends a backup routing reply message M_{br} back. The original node receives the reply message and establish the forward backup route. The process is expressed in Algorithm 5.

When a failure occurs in the data transmission process, the node that detects the failure chooses the node that wakes up the earliest in the multiple backup routes in the routing table as the next hop node. If multiple nodes wake up at the same time, the node with the smallest $d[v]$ will be chosen as the next hop. If $d[v]$ is still the same, select a node with a smaller sequence number.

Algorithm 5: Establish backup route

```

1: BEGIN
2: FOR each  $v \in V - \{\text{sink}\}$  DO
3:   Broadcast  $M_b$  with radius  $R_S$ 
4:   FOR each  $u \in V - \{v\}$  DO
5:     IF u receives  $M_b$  &&  $d[u] \leq d[v]$  && there is
6:       no reverse primary route from u to v DO
7:       u establish the reverse backup route to v
8:       u send  $M_{br}$  to v u along the reverse route
9:     END IF
10:   END FOR
11: IF v receives  $M_{br}$  DO
12:   v establish the forward backup route to u
13: END IF
14: END

```

After the backup routes are established, the WSN is shown in the Fig. 13.

5 Parameter optimization and performance analysis

5.1 Calculations of energy and SA slot

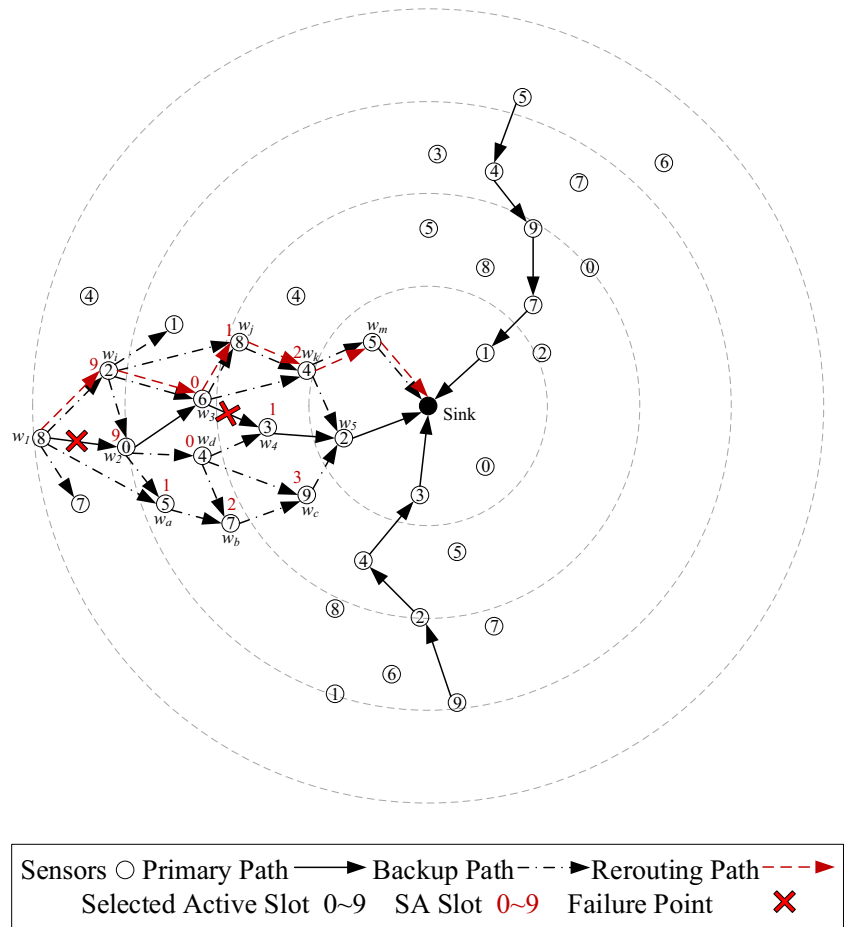
Definition 4 For simplicity, we define delay as the period from the next slot after the previous node receives the data to the slot before the latter node accepts the data packet.

In this paper, the node will switch to sleep state after completing the sending and receiving tasks in the active slot, so an active slot of a node can be divided into three parts: sending time, receiving time and sleep time. If SA slot is not added, then the node sleeps at other time of the cycle. If SA slot is added, then the node sends a beacon packet in SA slot to ask if there are packets that need to be transmitted. If there are data packets, it transmits them then, and if not, it switches into sleep state.

The biggest difference between this strategy and the previous ones is to add a SA slot using the energy of the peripheral nodes. Next we will analyze how far node away from the sink can afford a SA slot without decreasing network lifetime. Let the transmission rate of the network be \square , the length of a time slot be τ , the length of a working cycle be T and the duty cycle be $\varepsilon = \tau/T$. The following theorem can prove how far away from sink a SA slot can be added to the node without affecting the network lifetime. In our calculation, the energy consumption data of each state of the node is taken from Ref. [61], as shown in Table 5 below.

Theorem 1 In a network with radius R , the node transmission radius is r , event rate is λ , the node voltage is U , the size of data packets is ξ , the transmission rate of network

Fig. 6 A multi-failure rerouting under PSFR scheme



is \square , the time length of a slot is τ , and the length of a cycle is T . Then for each node whose distance from sink is greater than 2 hops, its residual energy is sufficient to add a SA slot.

Proof According to Ref. [59, 62], the amount of data borne by a node which is d hops away from sink can be expressed by Eq. (5).

Thus, we can work out that the energy consumption in a cycle of a node which is d hops away from sink without adding SA slot is as follows.

$$E_d = U(N_{d+1}\xi/u)A_{rx} + U(N_d\xi/u)A_{tx} + U(T - N_{d+1}\xi/u - N_d\xi/u)A_{sleep}$$

Table 4 Delay in a multi-failure rerouting

Scheme	RSR	PSFR
Delay(slot)	12	2

So, compared to the node with the highest energy consumption, the residual energy of a node which is d hops away from sink is shown below.

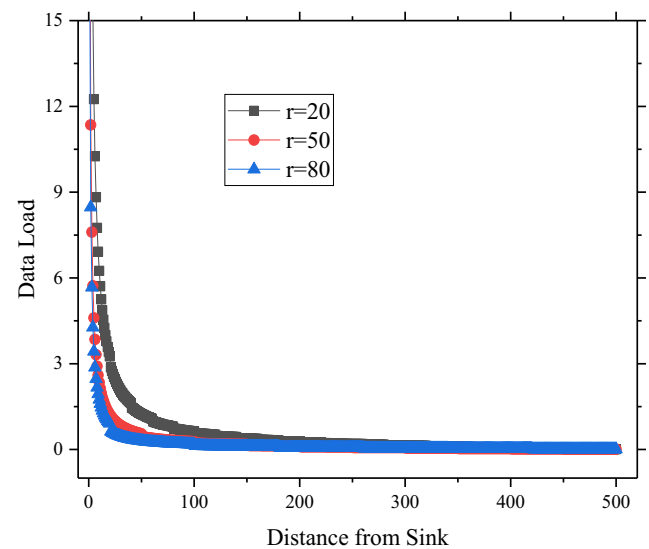


Fig. 7 The relation between the node's data load and its distance to sink

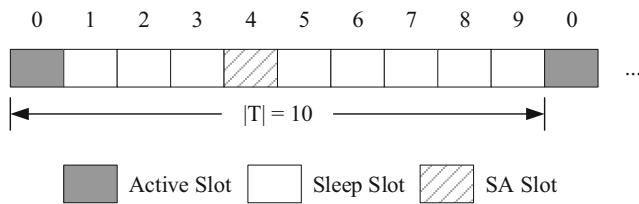


Fig. 8 The working cycle of a node with SA slot

$$\begin{aligned}
 E_{residual} &= E_1 - E_d \\
 &= \left(U(N_2\xi/u)A_{rx} + U(N_1\xi/u)A_{tx} \right. \\
 &\quad \left. + U(T - N_2\xi/u - N_1\xi/u)A_{sleep} \right) \\
 &\quad - \left(U(N_{d+1}\xi/u)A_{rx} + U(N_d\xi/u)A_{tx} \right. \\
 &\quad \left. + U(T - N_{d+1}\xi/u - N_d\xi/u)A_{sleep} \right) \\
 &= UA_{rx}\xi(N_2 - N_{d+1})/u \\
 &\quad + UA_{tx}\xi(N_1 - N_d)/u \\
 &\quad + UA_{sleep}\xi(N_{d+1} + N_d - N_2 - N_1)/u
 \end{aligned}$$

The increase in energy consumption by adding a SA slot (actually we only need to send a beacon) is calculated as follows. The size of a beacon s_{beacon} is 8 bytes (64bits), and the current of sending it is the same as the transmission current of ordinary data packets. So the energy consumption of adding a SA slot can be expressed by the following formula.

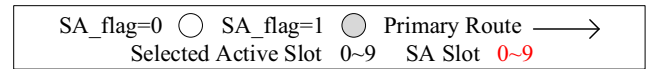
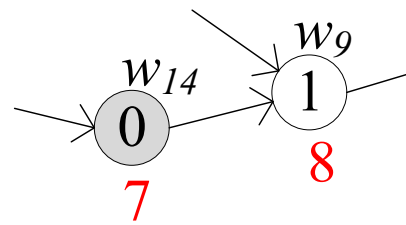


Fig. 10 The SA slot setting when that of the previous node has been set

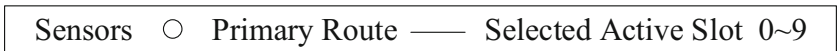
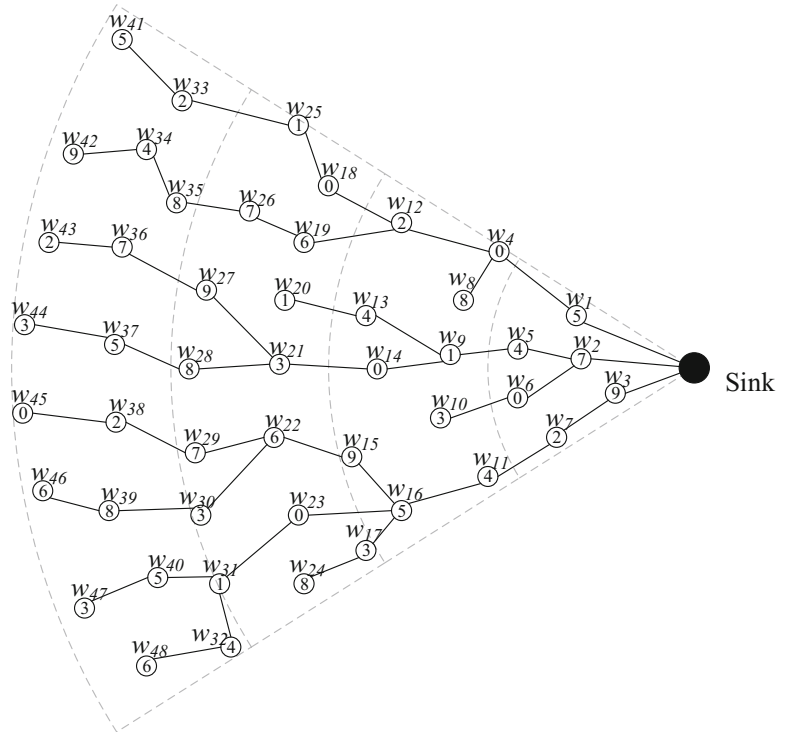
$$E_{beacon} = U(s_{beacon}/\Pi)A_{tx}$$

In WSNs, the size of ordinary data packet is much larger than that of beacon packet. As $N_1 - N_2 - s_{beacon}/\xi > 0$, we can deduce that $E_1 - E_2 \gg E_{beacon}$, which means the energy difference between the nodes with one hop away from sink and those with two hops away from sink is much more than the energy consumption of adding a SA slot.

Therefore, we can conclude that the nodes with more than 2 hops away from sink have enough residual energy to add a SA slot.

What follows is a specific example to illustrate Theorem 1. Suppose $T = 1s$, the duty cycle (the ratio of active slot to

Fig. 9 A WSN with primary routes established



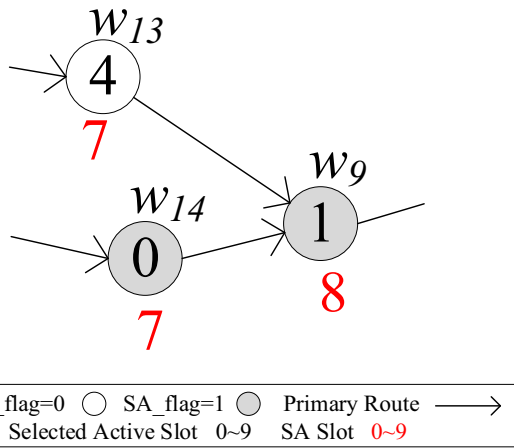


Fig. 11 The SA slot setting when that of the next node has been set

the whole cycle) ϵ is 0.5, the size of a data packet is 200 bytes, the data transmission rate is 200 kbps, the event rate $\mu = 0.01$, the network radius R is 500 m, the node transmission radius r is 80 m, the node voltage is 5 V, the network has 500 nodes and the size of beacon is 8 bytes. Then according to the formula in section 4.1, each node with one hop away from sink undertakes 16.87 data

packets, each node with two hops away from sink undertakes 8.47 data packets, and each of the peripheral nodes farthest from the center bears 0.01 data packets.

According to Table 3, if we add a SA slot to peripheral nodes, then sending a beacon packet consumes 0.01936 mJ, the energy consumption of the node with one hop from sink without adding a SA slot is 10.3179 mJ in one working cycle, the energy consumption of the node with two hops from sink with adding a SA slot is 5.5981 mJ in one working cycle, and 0.1152 mJ for the node with the farthest distance from sink node. The energy consumption of peripheral nodes with SA slot is much less than that of near sink nodes without SA slot, so it is feasible to add SA slot to peripheral nodes.

5.2 Delay calculation

5.2.1 Transmission delay in failure-free routing

Theorem 2 When SA slot is not added in the WSN, supposing the route length is k hops, the expectation of delay is $E_{DT} = (k(|T| - 1))/2$.

Fig. 12 A WSN with SA slot set

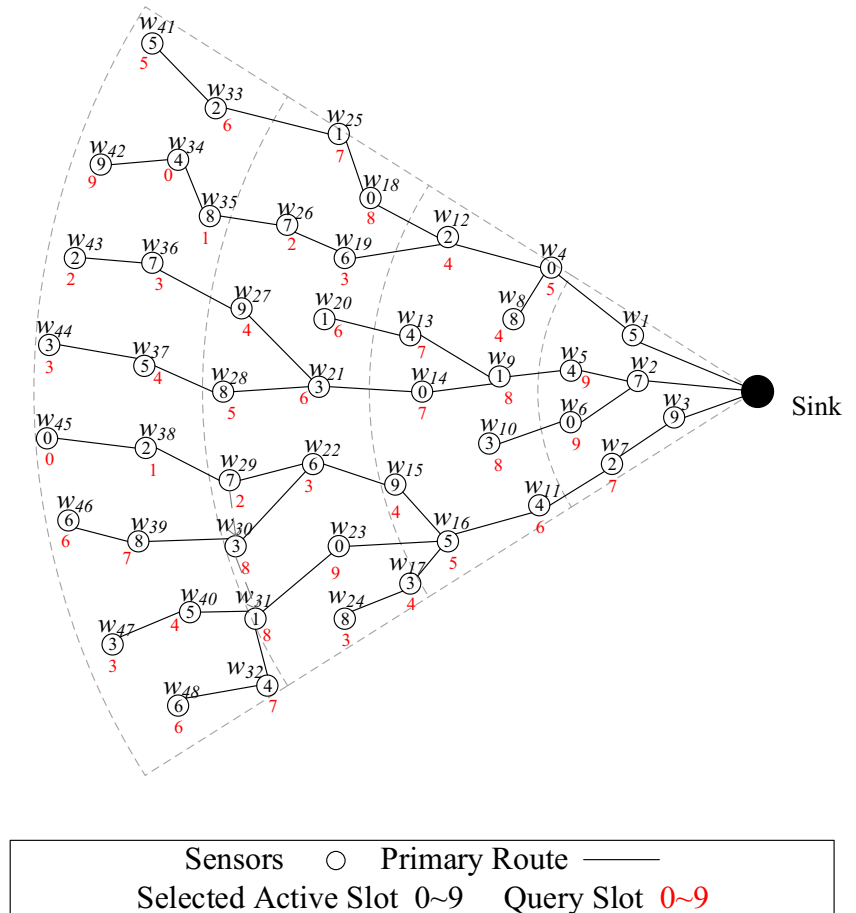
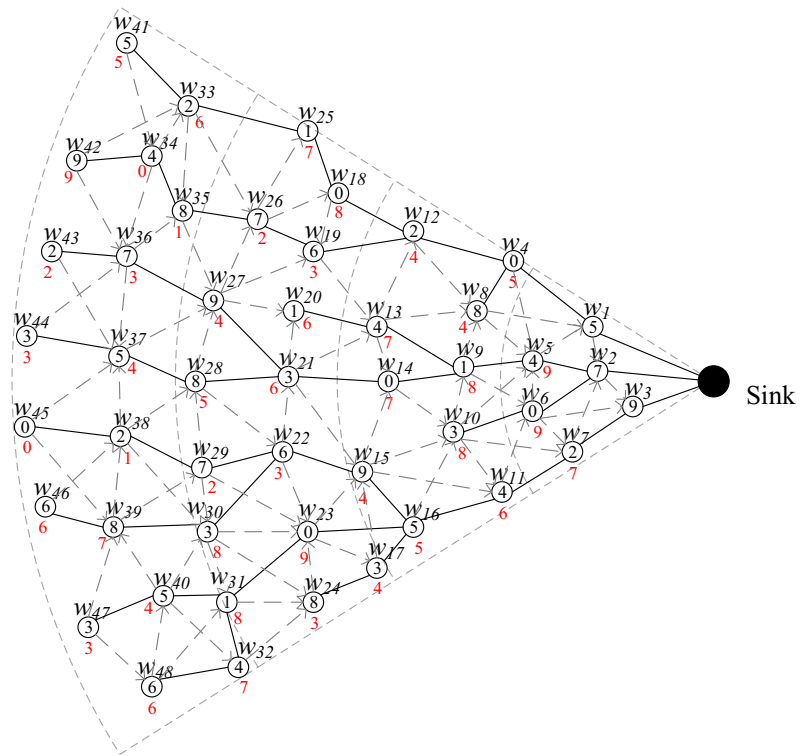


Fig. 13 A WSN with backup routes established



Sensors	○	Primary Route	——	Backup Route	- - -	→
		Selected Active Slot	0~9	Query Slot	0~9	

Proof Firstly, considering the delay of one hop, the probabilities of different delay lengths from node i to node j are as follows:

$$\begin{aligned}
 P(\text{delay} = 0) &= 1/|T|, \\
 P(\text{delay} = 1) &= 1/|T|, \\
 P(\text{delay} = 2) &= 1/|T|, \\
 &\vdots \\
 P(\text{delay} = |T|-1) &= 1/|T|.
 \end{aligned}$$

Therefore, the expectation of one-hop delay is $E_{D1} = 0 \times 1/|T| + 1 \times 1/|T| + 2 \times 1/|T| + \dots + (|T|-1) \times 1/|T| = (|T|-1)/2$.

Because the route length is k hops, the expectation of total delay is $E_{DT} = kE_{D1} = (k(|T|-1))/2$.

Table 5 Energy consumption in different states

Notation	Value	Description
A_{tx}	12.1 mA	Transmission current
A_{rx}	6.1 mA	Reception current
A_{sleep}	21.6 μ A	Sleep current

Theorem 3 When SA slot is added to peripheral nodes in the WSN, supposing the route length is k hops, the expectation of delay is $E_{DT} = (k-1)((5p+16)/96|T| - 5(p+2)/24|T| + 1/4) + (|T|-1)/2$.

Proof After adding SA slot, except that there is no SA slot near the sink node in one hop, the hop between other nodes can be divided into the following situations:

- (1) When the node that needs to transmit data received the message in SA slot (with a possibility of 1/2)
 - a) If the next hop node has no branches (with a possibility of 1-p): SA slot is continuous, and the delay is 0.
 - b) If the next hop node has branches (with a possibility of p): the SA slot may be continuous, in which case the delay is 0 (with a possibility of 1/2); the SA slot may be discontinuous, in which case the delay varies from 0 to $|T|/2$ (with a possibility of 1/2). Since E_m limits all branchings to bifurcations, the probability is 1/2.

When SA slot is discontinuous, the probabilities of delays of different lengths are as follows:

$$\begin{aligned}
 P(\text{delay} = 0) &= 1/((|T|/2) \times |T|/|T| + 1/|T| \\
 &\quad \times (|T|/2)/(|T|/2)-1/|T| \\
 &\quad \times 1/((|T|/2) = (3|T|-2)/|T|^2, \\
 P(\text{delay} = 1) &= 1/((|T|/2) \times (|T|-1)/|T| + 1/|T| \\
 &\quad \times (|T|/2-1)/(|T|/2)-1/|T| \\
 &\quad \times 1/((|T|/2) = (3|T|-6)/|T|^2, \\
 P(\text{delay} = 2) &= 1/((|T|/2) \times (|T|-2)/|T| + 1/|T| \\
 &\quad \times (|T|/2-2)/(|T|/2)-1/|T| \\
 &\quad \times 1/((|T|/2) = (3|T|-10)/|T|^2, \\
 &\vdots \\
 P(\text{delay} = |T|/2) &= 1/((|T|/2) \times (|T|-|T|/2)/|T| \\
 &\quad + 1/|T| \times (|T|/2-|T|/2)/(|T|/2) \\
 &\quad -1/|T| \times 1/((|T|/2) = (|T|-2)/|T|^2.
 \end{aligned}$$

So the delay expectation of the next hop when the node receives the message in the SA slot is $E_{SAS} = (1 - p) \times 0 + p \times (1/2 \times 0 + 1/2 \times ((3|T| - 2)/|T|^2 \times 0 + (3|T| - 6)/|T|^2 \times 1 + (3|T| - 10)/|T|^2 \times 2 + \dots + (|T| - 2)/|T|^2 \times |T|/2)) = (5p/2)(|T|/24 - 1/6|T|)$.

- (2) When the node that needs to transmit data received the message in active slot(with a possibility of 1/2), the next hop may wake up first in SA slot or in active slot.

$$\begin{aligned}
 P(\text{delay} = 0) &= 1/|T| \times |T|/|T| \times 2-1/|T| \times 1/|T| \\
 &= (2|T|-1)/|T|^2 \\
 P(\text{delay} = 1) &= 1/|T| \times (|T|-1)/|T| \times 2-1/|T| \\
 &\quad \times 1/|T| = (2|T|-3)/|T|^2 \\
 P(\text{delay} = 2) &= 1/|T| \times (|T|-2)/|T| \times 2-1/|T| \\
 &\quad \times 1/|T| = (2|T|-5)/|T|^2 \\
 &\vdots \\
 P(\text{delay} = |T|-1) &= 1/|T| \times (|T|-(|T|-1))/|T| \times 2 \\
 &\quad -1/|T| \times 1/|T| = 1/|T|^2
 \end{aligned}$$

So the delay expectation of the next hop when the node receives the message in the active slot is $E_{AS} = 0 \times (2|T| - 1)/|T|^2 + 1 \times (2|T| - 3)/|T|^2 + 2 \times (2|T| - 5)/|T|^2 + \dots + (|T| - 1) \times 1/|T|^2 = (|T| - 1)(2|T| + 5)/6|T|$.

Then the expectation of delay of this hop is the average of case (1) and case (2), which is $E_{D1} = (E_{SAS} + E_{AS})/2 = (5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4$.

Because the route is k hops long and the nodes within one hop distance of sink node does not set SA slot, the total delay expectation is $E_{DT} = (k - 1)E_{D1} + (|T| - 1)/2 = (k - 1)((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4) + (|T| - 1)/2$.

In order to display the average transmission delay that PSFR method deduced under different conditions according to the theorem and its optimization to the traditional RSR method, we take the network graph in Section 4.2 as an

example and use the formula from the theorem to calculate the average transmission delay of the network of different working cycle lengths with different hop number k . According to Theorem 2 and 3, taking the network graph established in Section 4.2 as an example, we can calculate the change of transmission delay length before and after adding SA slot for different $|T| = n$ values, which is shown in Fig. 14.

5.2.2 Transmission delay in single-failure routing

Theorem 4 When SA slot is not added in the WSN, supposing the original route length is k hops, a failure occurs at the s th hop and there is no other failures in the rerouting process, the expectation of delay is $E_{DT} = (2k + 1)(|T| - 1)/4$.

Proof Because the number of hops from sink of the next node on backup route is smaller than or the same as current node, and the backup routing request message M_b only covers nodes in one hop distance, the route length will change to k or $k + 1$ hops after a single failure. When there are enough nodes in the network, we think the probability of both cases is equal, that is, 1/2. From the proof of Theorem 1, the expectation delay of k -hops and $k + 1$ -hops are respectively $E_{Dk} = kE_{D1} = k(|T| - 1)/2$ and $E_{Dk+1} = (k + 1)E_{D1} = (k + 1)(|T| - 1)/2$.

So the total delay expectation is $E_{DT} = E_{Dk}/2 + E_{Dk+1}/2 = (2k + 1)(|T| - 1)/4$.

Theorem 5 When SA slot is added to peripheral nodes, supposing the original route length is k hops, a failure occurs at the s th hop and there is no other failures in the rerouting process, the expectation of delay is $E_{DT} = (k - 3/2)((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4) + (|T| - 1)/2 + (|T| - 1)(2|T| + 5)/24|T|$.

Proof Similar to the proof of Theorem 3, the probability of routing length changing to k or $k + 1$ hops is the same. According to the proof of Theorem 2, the expectation of delay of the $s - 1$ hops before failure, the $k - s$ hops after failure (if route length is still k) and $k - s + 1$ hops after failure (if route length becomes $k + 1$) are respectively:

$$E_{Ds-1} = (s-1)E_{D1} = (s-1) \left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4 \right),$$

$$E_{Dk-s} = (k-s-1)E_{D1} + (|T|-1)/2 = \left(k-s-1 \right) \left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4 \right) + (|T|-1)/2, \text{ and}$$

$$E_{Dk-s+1} = (k-s)E_{D1} + (|T|-1)/2 = \left(k-s \right) \left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4 \right) + (|T|-1)/2.$$

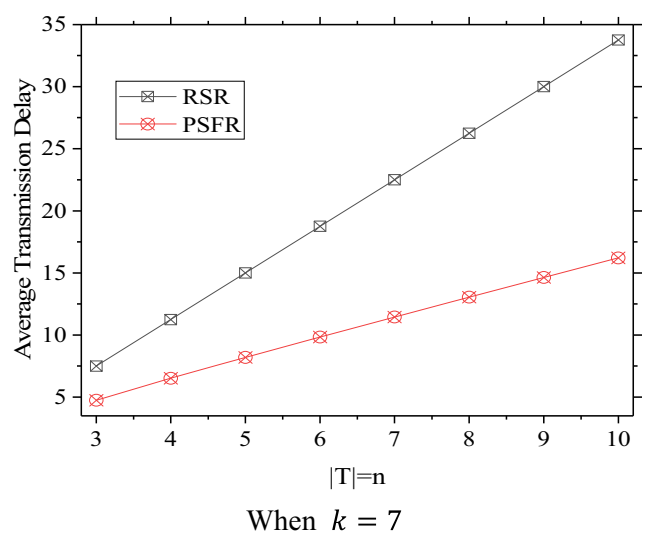
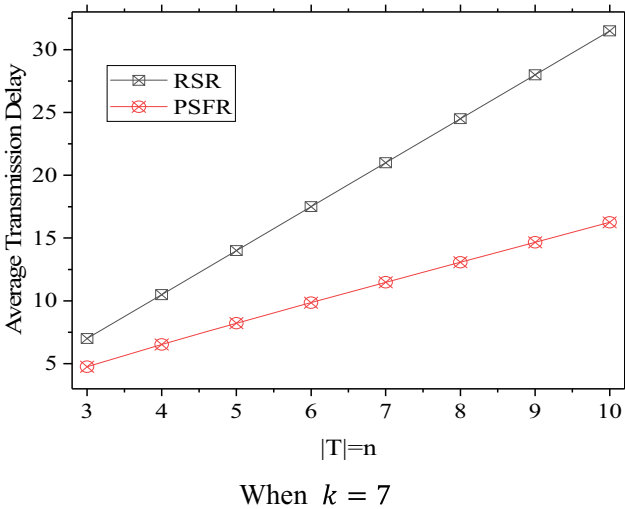
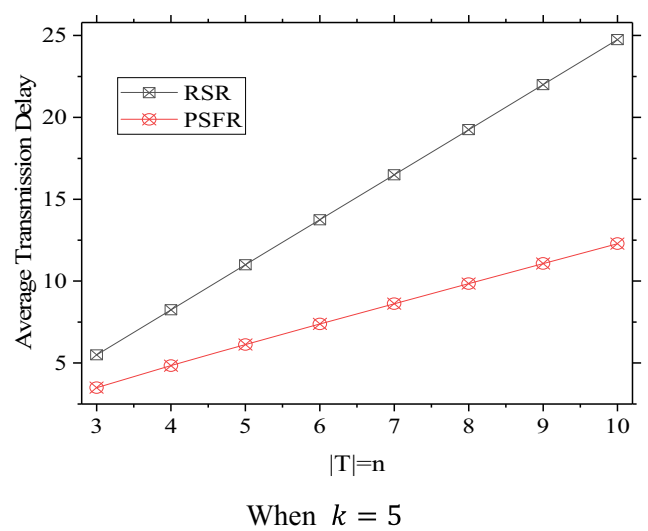
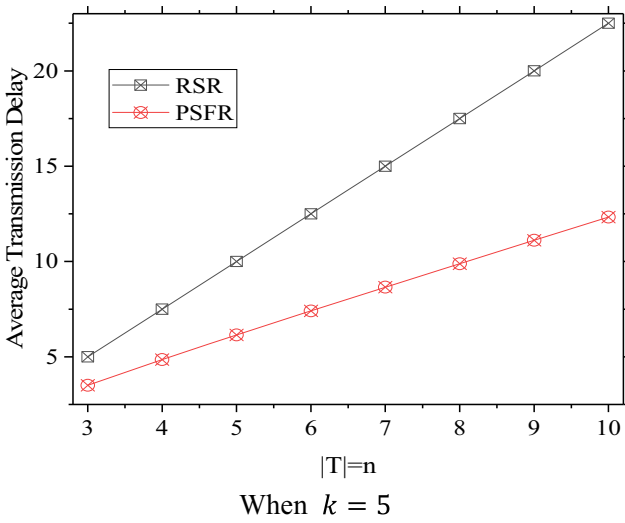
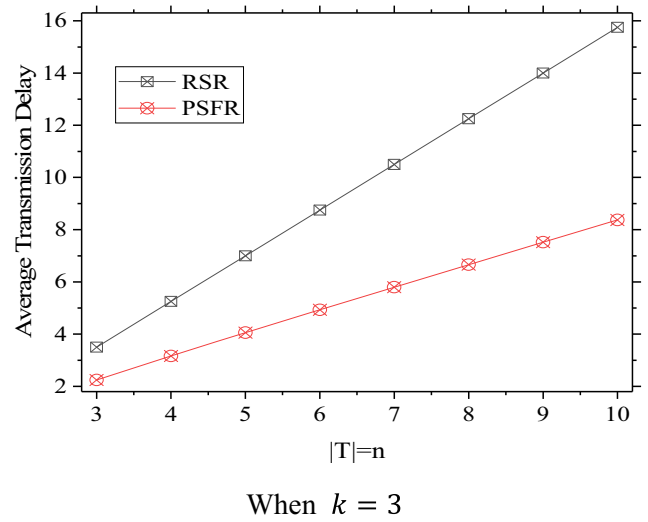
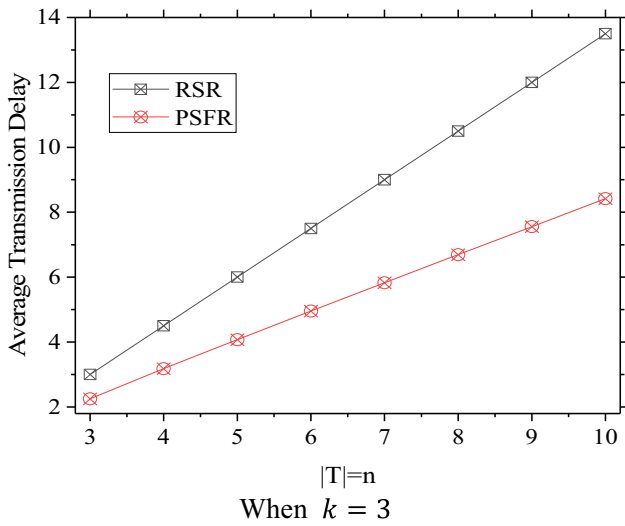


Fig. 14 The transmission delay before and after adding SA slot in failure-free case

Fig. 15 The transmission delay before and after adding SA slot in single-failure case

The delay expectation of the s th hop where a failure occurs is (assuming that each node has four backup routes of the next hop) $E_s = (|T| - 1)((2|T| + 5)/6|T|)/4 = (|T| - 1)((2|T| + 5)/24|T|)$.

Therefore, in the case of a single failure, the delay expectations of route length k and $k + 1$ are respectively:

$$E_{Dk} = E_{D_{s-1}} + E_s + E_{D_{k-s}}$$

$$= (k-2) \left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4 \right) + (|T|-1)/2 + (|T|-1)((2|T| + 5)/24|T|)$$

$$E_{D_{k+1}} = E_{D_{s-1}} + E_s + E_{D_{k-s+1}}$$

$$= (k-1) \left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4 \right) + (|T|-1)/2 + (|T|-1)((2|T| + 5)/24|T|)$$

So the total delay expectation is $E_{DT} = (E_{Dk} + E_{D_{k+1}})/2 = (k - 3/2)((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4) + (|T| - 1)/2 + (|T| - 1)((2|T| + 5)/24|T|)$.

According to Theorem 4 and Theorem 5, taking the network graph established in Section 4.2 as an example, we can calculate the change of transmission delay length before and after adding SA slot for different $|T| = n$ values, which is shown in Fig. 15.

5.2.3 Transmission delay in multi-failure routing

Theorem 6 When SA slot is not added in the WSN, supposing the original route length is k hops and there are m failure arbitrarily occur in the routing process, the expectation of delay is $E_{DT} = (|T| - 1)(k + m/2)/2$.

Proof When backup route is selected after each failure, the length of routing either increases by one hop or does not increase. Therefore, the original k -hop routing length becomes $k \sim k + m$ hops when m failures occur at any times. According to Theorem 1, the delay expectation of $k + m$ hops is $E_{D_{k+m}} = (k + m)E_{D1} = (k + m)(|T| - 1)/2$. The probabilities of increasing different numbers of hops are as follows.

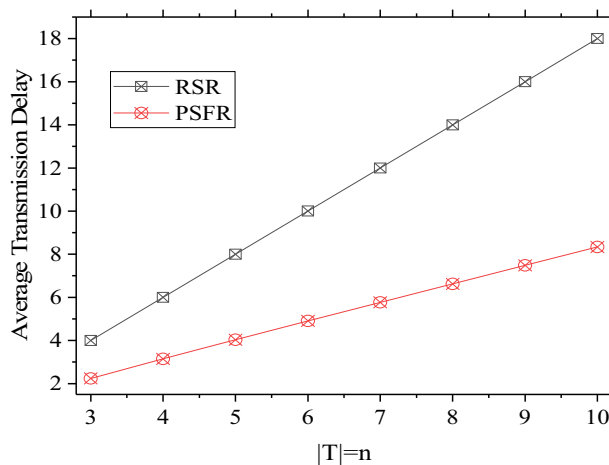
$$P(H_N = k) = C_m^0 (1/2)^m$$

$$P(H_N = k + 1) = C_m^1 (1/2)^m$$

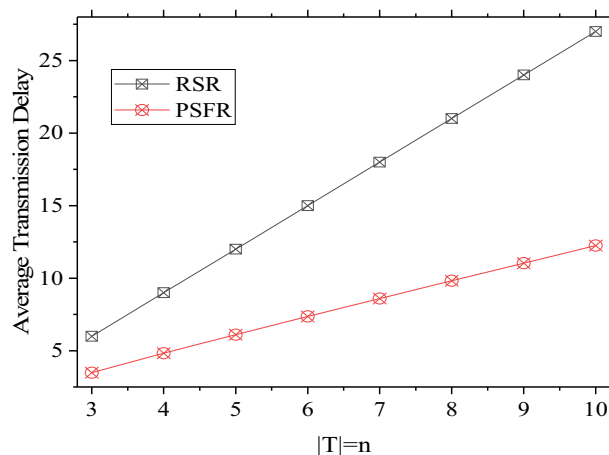
$$P(H_N = k + 2) = C_m^2 (1/2)^m$$

$$\vdots$$

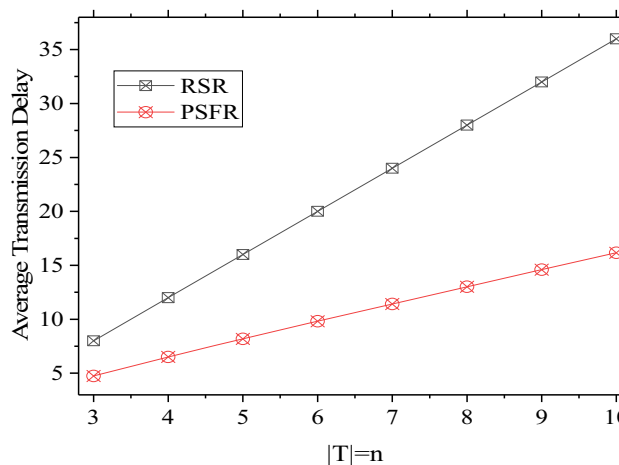
$$P(H_N = k + m) = C_m^m (1/2)^m$$



When $k = 3, m = 2$



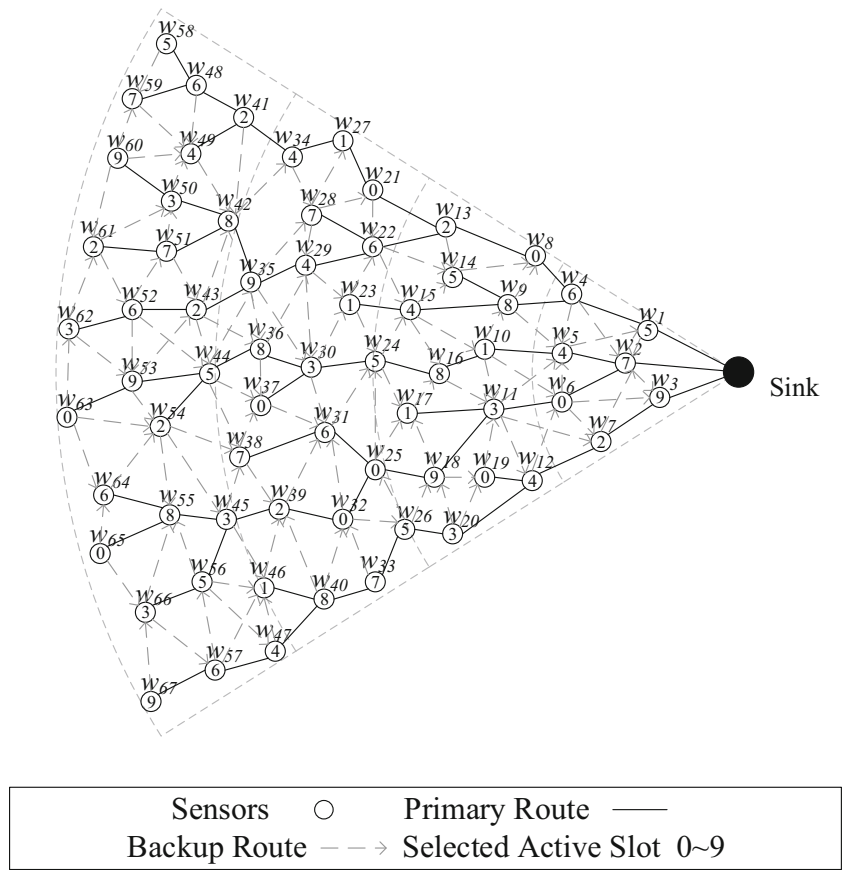
When $k = 5, m = 2$



When $k = 7, m = 2$

Fig. 16 The transmission delay before and after adding SA slot in multi-failure case

Fig. 17 Network 1(RSR scheme)



The total delay expectation is $E_{DT} = C_m^0(1/2)^m E_{Dk} + C_m^1(1/2)^m E_{Dk+1} + C_m^2(1/2)^m E_{Dk} + 2 + \dots + C_m^m(1/2)^m E_{Dk+m} = (|T|-1)(k + m/2)/2$.

Theorem 7 When SA slot is added to peripheral nodes, supposing the original route length is k hops and there are m failure arbitrarily occur in the routing process, the expectation of transmission delay is $E_{DT} = (k - m/2 - 1)((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4) + (|T| - 1)/2 + m(|T| - 1)((2|T| + 5)/24|T|)$.

Proof Similar to the proof of Theorem 5, the route length becomes $k-k+m$ hop, and the probabilities of different route length are the same as those in Theorem 5.

For the new routes of different lengths, the delays expectations of the failure-free part are:

$$\begin{aligned}
 E_{Dk} &= (k-1)E_{D1} + (|T|-1)/2 \\
 &= (k-1)\left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4\right) \\
 &\quad + (|T|-1)/2 \\
 E_{Dk-1} &= (k-2)E_{D1} + (|T|-1)/2 \\
 &= (k-2)\left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4\right) \\
 &\quad + (|T|-1)/2 \\
 E_{Dk-2} &= (k-3)E_{D1} + (|T|-1)/2 \\
 &= (k-3)\left((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4\right) \\
 &\quad + (|T|-1)/2 \\
 &\vdots
 \end{aligned}$$

$$E_{Dk-m} = (k - m - 1)E_{D1} + (|T| - 1)/2 = (k - m - 1)((5p + 16)|T|/96 - 5(p + 2)/24|T| + 1/4) + (|T| - 1)/2.$$

Fig. 18 Network 1(PSFR scheme)

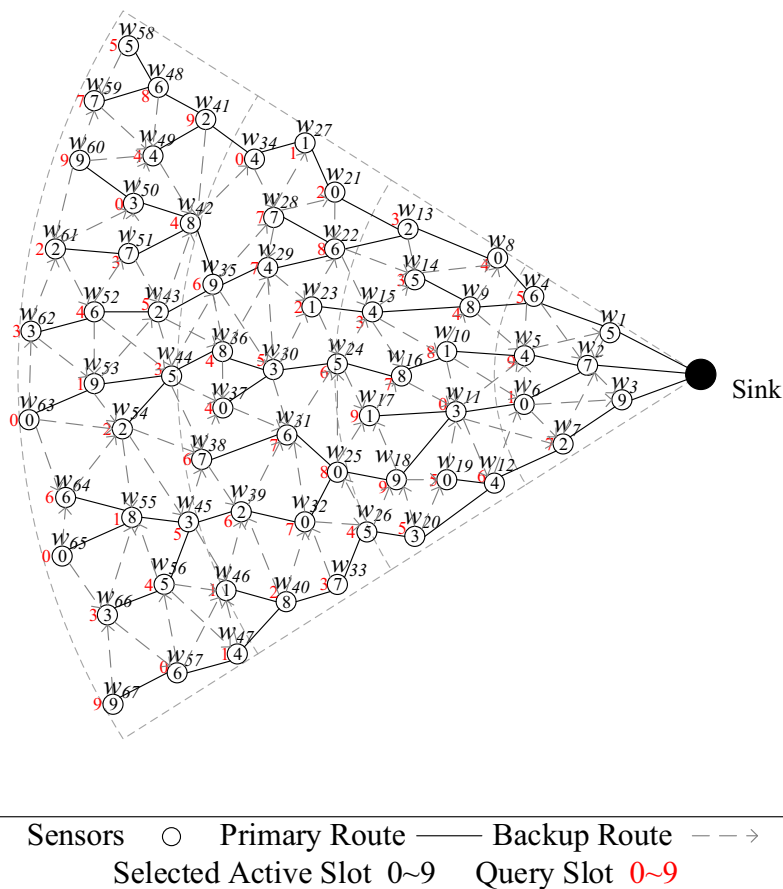


Figure 18. Network 1(PSFR Scheme)

The delay expectation of the sth hop where the failure occurs is (assuming that each node has four backup routes of the next hop) $E_s = (|T| - 1)((2|T| + 5)/6|T|)/4 = (|T| - 1)((2|T| + 5)/24|T|)$.

Therefore, in the case of multiple failures, the delay expectation values of route length $k \sim k + m$ are respectively:

$$E_{Dk} = E_{Dk-m} + mE_s = (k-m-1) \left(\frac{(5p+16)|T|}{96} - \frac{5(p+2)}{24|T|} + \frac{1}{4} \right) + \frac{(|T|-1)}{2} + m \frac{(|T|-1)((2|T|+5)}{24|T|}$$

$$E_{Dk+1} = E_{Dk-m+1} + mE_s = (k-m) \left(\frac{(5p+16)|T|}{96} - \frac{5(p+2)}{24|T|} + \frac{1}{4} \right) + \frac{(|T|-1)}{2} + m \frac{(|T|-1)((2|T|+5)}{24|T|}$$

$$E_{Dk+2} = E_{Dk-m+2} + mE_s = (k-m+1) \left(\frac{(5p+16)|T|}{96} - \frac{5(p+2)}{24|T|} + \frac{1}{4} \right) + \frac{(|T|-1)}{2} + m \frac{(|T|-1)((2|T|+5)}{24|T|}$$

⋮

$$E_{Dk+m} = E_{Dk} + mE_s = (k-1) \left(\frac{(5p+16)|T|}{96} - \frac{5(p+2)}{24|T|} + \frac{1}{4} \right) + \frac{(|T|-1)}{2} + m \frac{(|T|-1)((2|T|+5)}{24|T|}$$

Therefore, the total delay expectation is

$$E_{DT} = (E_{Dk} + E_{Dk+1} + E_{Dk+2} + \dots + E_{Dk+m})/m + 1 = \left(k - \frac{m}{2} - 1 \right) \left(\frac{(5p+16)|T|}{96} - \frac{5(p+2)}{24|T|} + \frac{1}{4} \right) + \frac{(|T|-1)}{2} + m \frac{(|T|-1)((2|T|+5)}{24|T|}$$

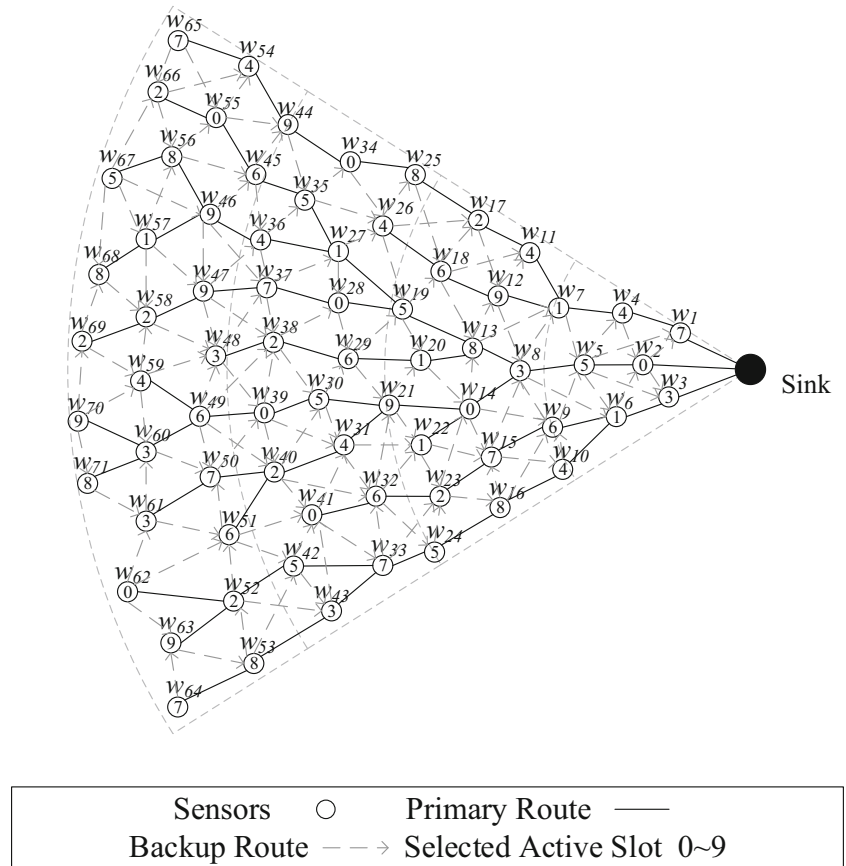
According to Theorem 6 and Theorem 7, taking the network graph established in Section 4.2 as an example, we can work out the change of transmission delay length in the network before and after adding SA slot to the peripheral nodes for different $|T| = n$ values, which is shown in Fig. 16.

6 Analysis of experimental results

6.1 Experiment setup

Experimental scenario is as follows:

Fig. 19 Network 2(RSR scheme)



In the experiment, we use two randomly generated network graphs to compare the two schemes (RSR and SFR) in terms of induction delay, transmission delay, network lifetime and rerouting delay. The first network has one sink node and 67 ordinary nodes, and the second network has one sink node and 71 ordinary nodes.

Figure 17 shows the case when Network 1 adopts the RSR scheme and establishes the primary route and backup route without adding SA slots.

After adopting the PSFR scheme, SA slots are added to Network 1, as is shown in Fig. 18.

Similarly, when Network 2 adopts the RSR scheme, it establishes the primary route and backup route as Fig. 19 shows.

After adopting the PSFR scheme and adding SA slots, Network 2 turns out to be what Fig. 20 shows.

The parameters for energy consumption calculation are given in Table 6.

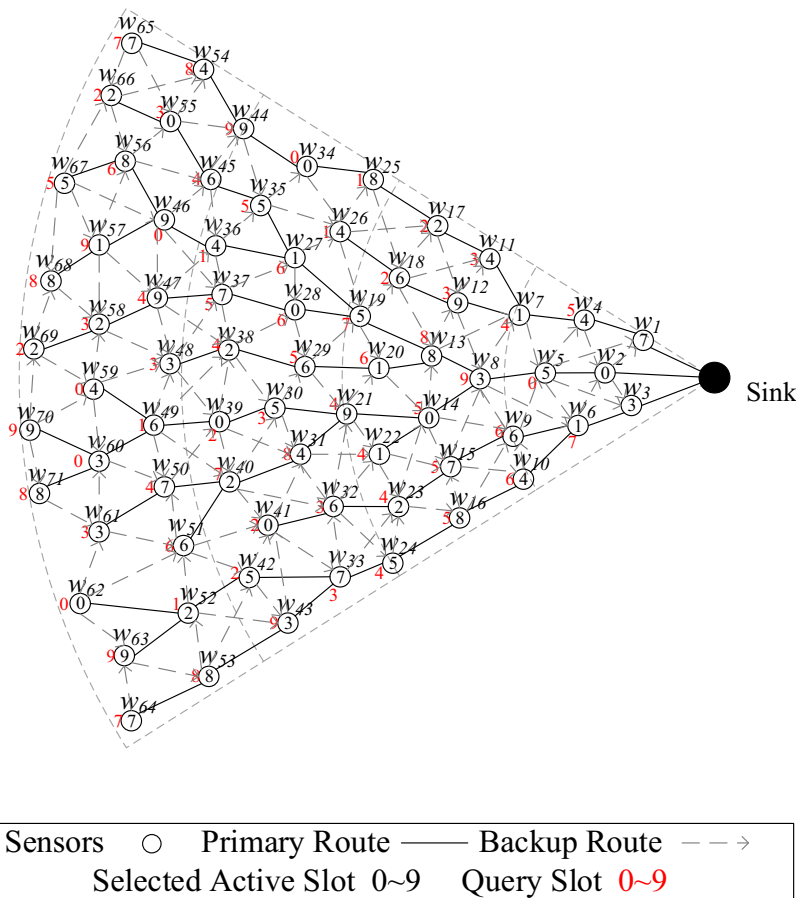
6.2 Results and discussion

6.2.1 Sensing delay

We define the sensing delay as the average delay from the occurrence of an event to the detection of it by the first node.

Figure 21 shows the sensing delays of RSR and PSFR scheme in Network 1 and Network 2, where events are generated at random time and place. It can be seen that in Network 1, the sensing delay using PSFR scheme is 49.42% less than that using RSR scheme; in Network 2, the sensing delay using PSFR scheme is 56.55% less than that using RSR scheme. It can be concluded that in PSFR scheme, the nodes in the network can detect the events happening in the surrounding environment more quickly, and the average performance is much better than that of RSR scheme.

Fig. 20 Network 2(PSFR scheme)



6.2.2 Transmission delay

We define transmission delay as the average value of delay from the node to sink after the node detects an event in the network as the first node in a failure-free situation.

In the RSR scheme, the transmission delay is the delay from the active slot of the node receiving the data to sink. In

the PSFR scheme, the transmission delay is the average value of the delay starting from the original active slot of the node receiving the data and from the SA slot.

In the two networks, when the first node detected event nodes is w_{20} , w_{50} , w_{60} , w_{25} , w_{55} and w_{65} , the transmission delay using RSR scheme and PSFR scheme is compared in Fig. 22.

Table 6 Parameters

Notation	Value
Duty cycle ϵ	0.1
Working cycle T	0.1 s
Event rate μ	0.01
Network radius R	10 m
Node transmission radius r	1 m
Data transmission rate ν	200 kpbs
Data packet size s_{data}	200 bytes
Beacon package size s_{beacon}	8 bytes
Initial energy of each node e_{mit}	1000 mJ
Node voltage U	5 V
Receiving current A_{rx}	6.1 mA
Transmit current A_{tx}	12.1 mA
Sleep current A_{sleep}	21.6 μ A

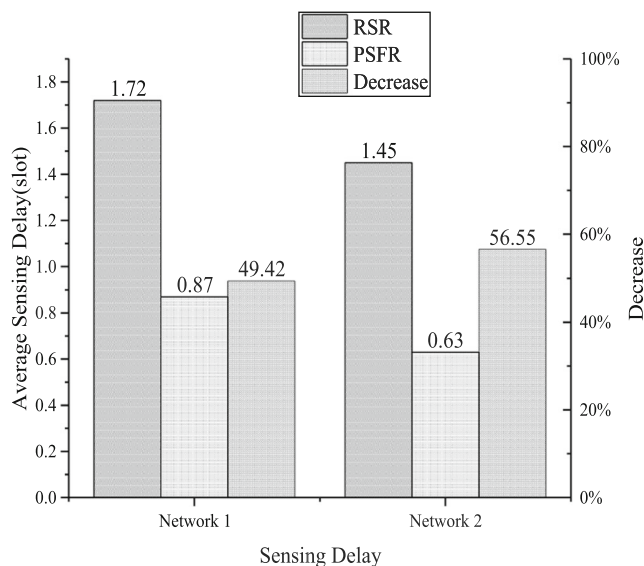


Fig. 21 Average sensing delay

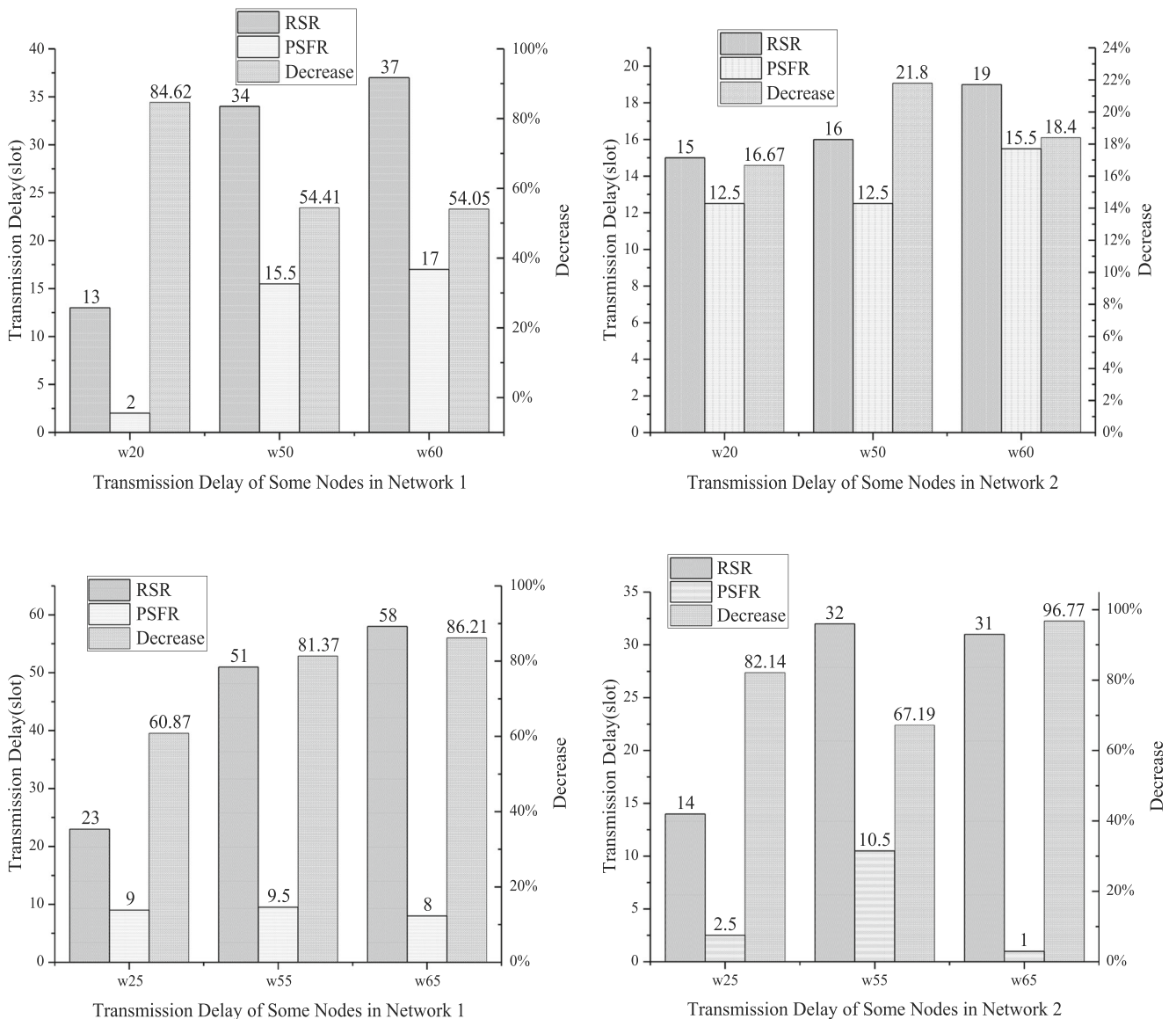


Fig. 22 Transmission delay of some nodes

After calculating the transmission delays of all the nodes in the network, the average transmission delays of the two networks using the two schemes are obtained, and the effect comparison is shown in Fig. 23.

It can be seen that after adopting PSFR scheme, the transmission delay from each node to sink in network one is 67.18% lower than that adopting RSR scheme, and the transmission delay from each node to sink in network two is 49.25% lower than that adopting RSR scheme, which greatly improves the transmission efficiency. Obviously, the performance of PSFR in minimizing transmission delay is much better than that of RSR, and the waiting time for data packet transmission in network is significantly reduced.

6.2.3 Energy utilization rate

In WSNs, network lifetime is determined by the earliest energy-drained-out nodes. The nodes nearest to the sink carries the largest amount of data, while the initial energy of all ordinary nodes is the same, so the nodes nearest to sink consumes energy at the fastest rate, and their lifetime is the lifetime of the network.

According to the experimental parameters set in Section 6.1, we can work out that the lifetime of the network is 292.25 s. When RSR method is adopted, the peripheral node will go to sleep after completing the work of sending and receiving data packets in the active slot of each cycle. In PSFR method, besides the normal active slot activity, the

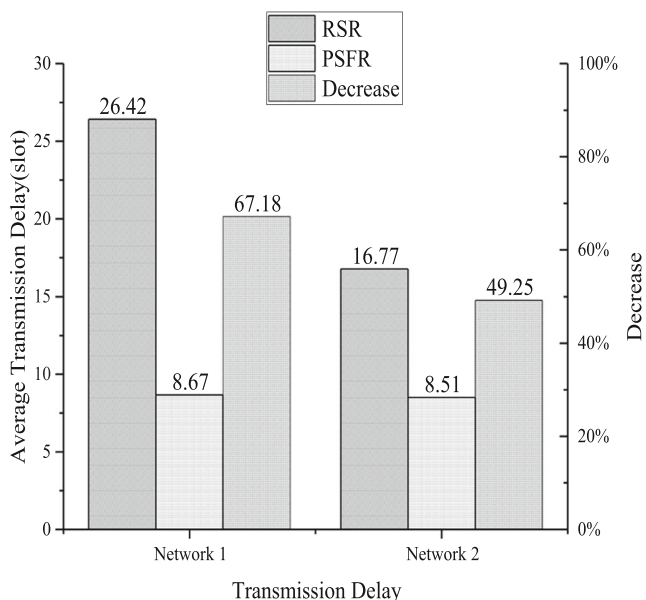


Fig. 23 Average transmission delay

peripheral node will also send data in the SA slot. A beacon packet asks if there is data to be sent. Since the power of sending beacon packets is higher than that of the nodes in sleep, networks adopting the PSFR scheme consume more energy than adopting the RSR scheme.

In the RSR scheme, the energy consumption of nodes with different hops away from sink in one cycle T is respectively shown in Table 7.

In the PSFR scheme, due to the addition of SA slot, the energy consumption of nodes with different hops from sink in one cycle T becomes as follows in Table 8.

From these two tables, we can calculate the energy utilization of RSR and PSFR in the two networks, which is shown in Fig. 24.

We can see that in Network 1, the energy utilization rate of PSFR is 27.79% higher than that of RSR, and in Network 2, the energy utilization rate of PSFR is 27.53% higher than that of RSR. Obviously, the PSFR proposed in this paper improves

Table 7 Energy consumption of nodes in RSR scheme

Hops away from sink	Energy consumption in one working cycle
1	0.3421 mJ
2	0.1826 mJ
3	0.1221 mJ
4	0.0907 mJ
5	0.0713 mJ
6	0.0568 mJ
7	0.0423 mJ
8	0.0302 mJ
9	0.0229 mJ
10	0.0156 mJ

Table 8 Energy consumption of nodes in PSFR scheme

Hops away from sink	Energy consumption in one working cycle
1	0.3421 mJ
2	0.2019 mJ
3	0.1414 mJ
4	0.1100 mJ
5	0.0906 mJ
6	0.0761 mJ
7	0.0616 mJ
8	0.0495 mJ
9	0.0422 mJ
10	0.0350 mJ

the utilization of energy in the network without affecting the network lifetime. Compared with RSR, PSFR improves other aspects of network performance while maintaining the maximum network lifetime.

6.2.4 Rerouting delay

We define a rerouting delay as the average delay of a data packet to sink after a failure occurs on the primary route.

In the two networks, when the first node that detects a failure on the next hop of the primary route is w_{20} , w_{50} , w_{60} , w_{25} , w_{55} and w_{65} , the rerouting delay using RSR scheme and PSFR scheme is compared in Fig. 25.

After calculating all the nodes in the two networks, we can get the average single-failure rerouting delay using RSR and PSFR scheme as shown in Fig. 26.

It can be seen that the rerouting delay from each node to sink in Network 1 is 62.28% lower than that of RSR, and the rerouting delay from each node to sink in Network 2 is

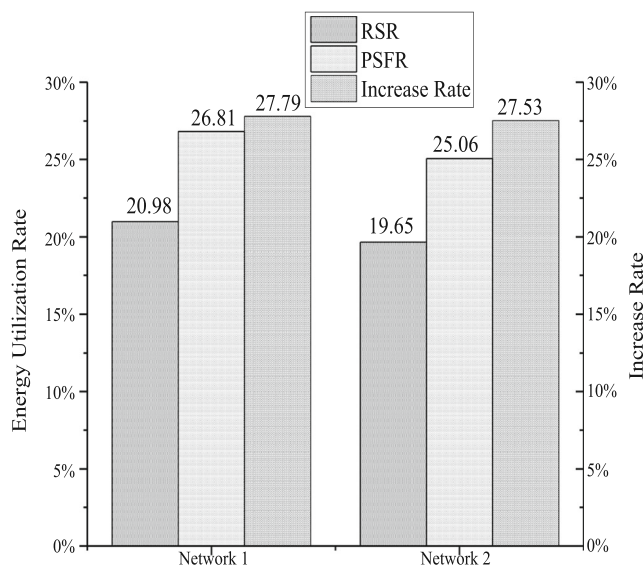


Fig. 24 Network energy utilization rate

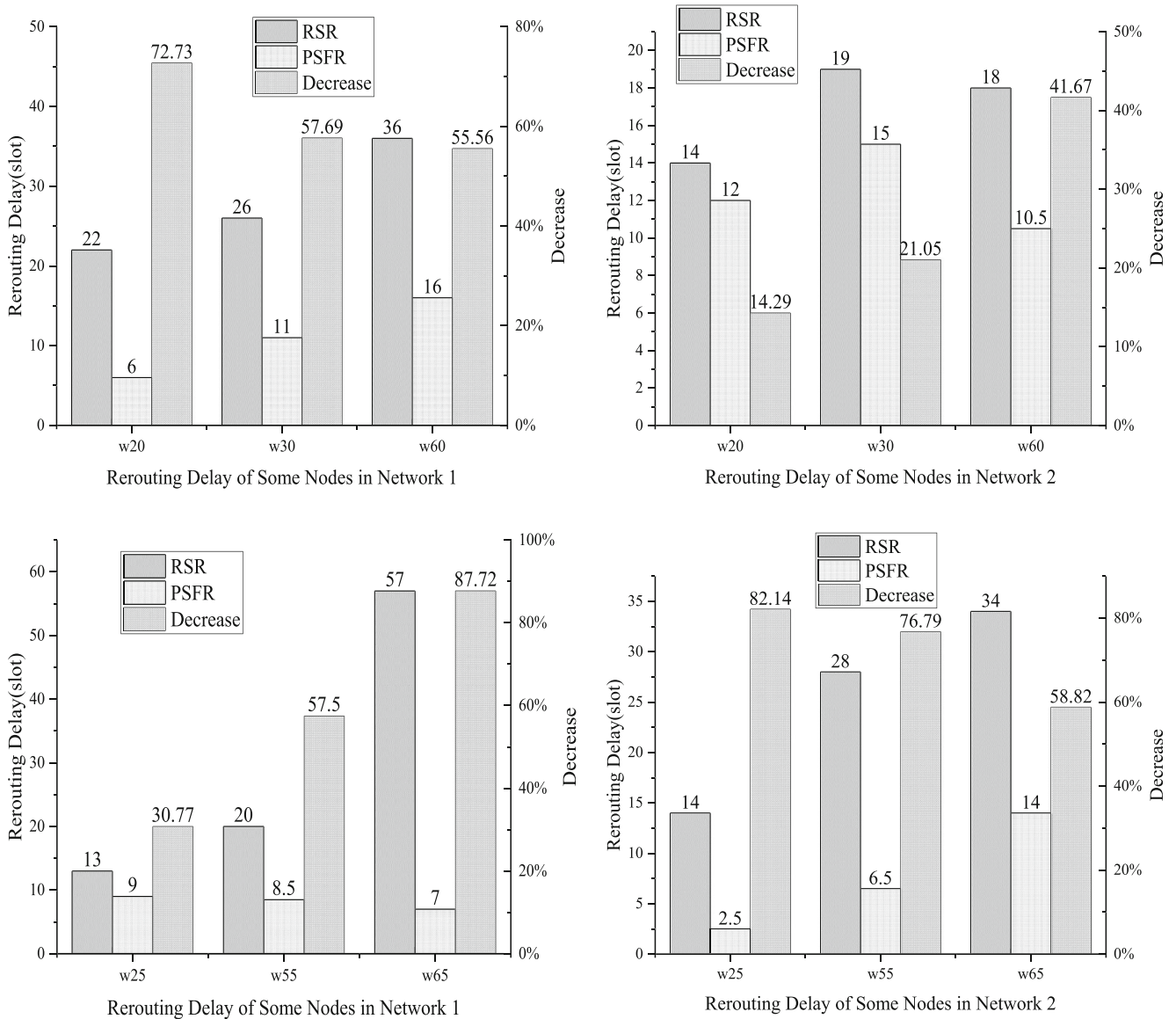


Fig. 25 Rerouting delay of some nodes

48.23% lower than that of RSR, which greatly improves the transmission efficiency.

Obviously, PSFR is much better than RSR in reducing the rerouting delay. Compared with RSR scheme, PSFR scheme can reduce the single-failure rerouting delay by about half on average. When the data transmission path fails, the rerouting time in PEFR is significantly less than the traditional RSR scheme.

7 Conclusion

In circular WSNs with sink nodes as the center, the traditional RSR scheme for data transmission has the shortcomings of discontinuous slots in data transmission and the need for

nodes to wait for the next hop node to wake up to transmit data, resulting in a large delay. Besides, the residual energy of peripheral nodes can not be utilized after network death. To overcome the shortcomings of this method, we do the research and propose a PSFR scheme, which uses the residual energy of peripheral nodes to add a sequential active (SA) slot to the next slot of that when the last node wakes up, so that the data packets can be transmitted continuously and the delay of waiting for the next node to wake up is reduced. In backup routing, SA Slot also reduces the delay of data packet rerouting with relatively loose settings. Adding a SA Slot will increase the energy consumption of nodes, but because the data load of nodes near the center is large and the energy is consumed fast, the network lifetime depends on the lifetime of nodes near the center. Therefore, Adding a SA Slot on

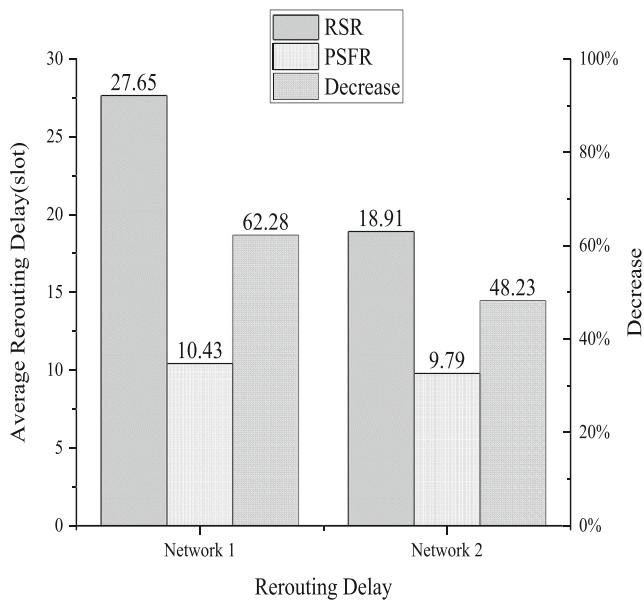


Fig. 26 Average rerouting delay

peripheral nodes that have a large amount of energy surplus will not reduce network lifetime. The improved WSNs will perform better in applications of the physical world.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (61772554), Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (NLBKF201804) and Natural Science Foundation of Zhejiang Province (LY17F020032).

References

- Riaz S, Rehan M, Umer T, Afzal MK, Rehan W et al (2018) FRP: A novel fast rerouting protocol with multi-link-failure recovery for mission-critical WSN. *Futur Gener Comput Syst* 89(2018):148–165
- Wang T, Zhang G, Bhuiyan M, Liu A, Jia W, Xie M (2018) A novel trust mechanism based on fog computing in sensor-cloud system. *Futur Gener Comput Syst*. <https://doi.org/10.1016/j.future.2018.05.049>
- Liu Y, Liu A, Xiong N, Wang T, Gui W (2019) Content propagation for content-centric networking from location-based social networks. *IEEE Trans Syst Man Cybern Syst*. <https://doi.org/10.1109/TSMC.2019.2898982>
- Yang C, Shi Z, Han K, Zhang J, Gu Y, Qin Z (2018) Optimization of particle CBMeMber filters for hardware implementation. *IEEE Trans Veh Technol* 67(9):9027–9031
- Huang B, Liu W, Wang T, Li X, Song H, Liu A (2019) Deployment optimization for data centers in vehicular networks. *IEEE Access* 7(1):20644–20663
- Li Z, Liu Y, Liu A, Wang S, Liu H (2018) Minimizing convergecast time and energy consumption in green internet of things. *IEEE Trans Emerg Top Comput*. <https://doi.org/10.1109/TETC.2018.2844282>
- Zhang C, Chen R, Zhu L, Liu A, Lin Y, Huang F (2018) Hierarchical information quadtree: Efficient spatial temporal image search for multimedia stream. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-018-6284-y>
- Huang M, Liu A, Xiong N et al (2019) A low-latency communication scheme for mobile wireless sensor control systems. *IEEE Trans Syst Man Cybern Syst* 49(2):317–332
- Zhang S, Li X, Tan Z, Peng T, Wang G (2019) A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services. *Futur Gener Comput Syst* 94:40–50
- Shi Z, Zhou C, Gu Y et al (2017) Source estimation using coprime array: A sparse reconstruction perspective. *IEEE Sensors J* 17(3):755–765
- Ren Y, Liu W, Wang T, Li X, Xiong N, Liu A (2019) A collaboration platform for effective task and data reporter selection in crowdsourcing network. *IEEE Access* 7(1):19238–19257
- Deng X, Luo J, He L, Liu Q, Li X, Cai L (2019) Cooperative channel allocation and scheduling in multi-interface wireless mesh networks. *Peer-to-Peer Networking and Applications* 12(1):1–12
- Zhang H, Zheng W (2018) Denial-of-service power dispatch against linear quadratic control via a Fading Channel. *IEEE Trans Autom Control* 63(9):3032–3039
- Liu X, Liu Y, Zhang N, Wu W, Liu A (2019) Optimizing trajectory of unmanned aerial vehicles for efficient data acquisition: A matrix completion approach. *IEEE Internet Things J*. 6(2):1829–1840. <https://doi.org/10.1109/JIOT.2019.2894257>
- Liu Y, Liu A, Liu X, Huang X (2019) A statistical approach to participant selection in location-based social networks for offline event marketing. *Inf Sci* 480:90–108
- Xiang X, Liu W, Xiong N, Song H, Liu A, Wang T (2018) Duty cycle adaptive adjustment based device to device (D2D) communication scheme for WSNs. *IEEE Access* 6(1):76339–76373
- Zhou C, Gu Y, He S et al (2018) A robust and efficient algorithm for coprime array adaptive beamforming. *IEEE Trans Veh Technol* 67(2):1099–1112
- Tan J, Liu W, Xie M, Song H, Liu A, Zhao M, Zhang G (2019) A low redundancy data collection scheme to maximize lifetime using matrix completion technique. *EURASIP J Wirel Commun Netw* 2019:5. <https://doi.org/10.1186/s13638-018-1313-0>
- Li T, Tian S, Liu A, Liu H, Pei T (2018) DDSV: Optimizing delay and delivery ratio for multimedia big data collection in mobile sensing vehicles. *IEEE Internet Things J* 5(5):3474–3486
- Ju X, Liu W, Zhang C, Liu A, Wang T, Xiong N, Cai Z (2018) An energy conserving and transmission radius adaptive scheme to optimize performance of energy harvesting sensor networks. *Sensors* 18(9):2885. <https://doi.org/10.3390/s18092885>
- Yang W, Liu W, Zeng Z, Liu A, Huang G, Xiong N, Cai Z (2018) Adding active slot joint larger broadcast radius for fast code dissemination in WSNs. *Sensors* 18(11):4055. <https://doi.org/10.3390/s18114055>
- Qi W, Liu W, Liu X, Liu A, Wang T, Xiong N, Cai Z (2018) Minimizing delay and transmission times with Long lifetime in code dissemination scheme for high loss ratio and low duty cycle WSNs. *Sensors* 18(10):3516. <https://doi.org/10.3390/s18103516>
- Chen Z, Liu A, Li Z, Choi Y, Li J (2018) Distributed duty cycle control for delay improvement in wireless sensor networks. *Peer Peer Netw Appl* 10(3):559–578
- Naveen KP, Kumar A (2013) Relay selection for geographical forwarding in sleep-wake cycling wireless sensor networks. *IEEE Trans Mob Comput* 12(3):475–488
- Teng H, Liu Y, Liu A, Xiong NN, Cai Z, Wang T, Liu X (2019) A novel code data dissemination scheme for internet of things through Mobile vehicle of smart cities. *Futur Gener Comput Syst* 94:351–367
- Ren J, Guo H, Xu C, Zhang Y (2017) Serving at the edge: A scalable IoT architecture based on transparent computing. *IEEE Netw* 31(5):96–105
- Yang G, He S, Shi Z (2017) Leveraging crowdsourcing for efficient malicious users detection in large-scale social networks. *IEEE Internet Things J* 4(2):330–339

28. Zhu Y, Zhong Z, Basin M, Zhou D (2018) A descriptor system approach to stability and stabilization of discrete-time switched PWA systems. *IEEE Trans Autom Control* 63(10):3456–3463
29. Yang G, He S, Shi Z, Chen J (2017) Promoting cooperation by social incentive mechanism in mobile crowdsensing. *IEEE Commun Mag* 55(3):86–92
30. Zhou H, Wang H, Chen X, Li X, Xu S (2018) Data offloading techniques through vehicular ad hoc networks: A survey. *IEEE Access* 6(1):65250–65259
31. Zhu Y, Zhong Z, Zheng W, Zhou D (2018) HMM-based H-infinity filtering for discrete-time Markov jump LPV systems over unreliable communication channels. *IEEE Trans Syst Man Cybern Syst* 48(12):2035–2046
32. Chen J, Hu K, Wang Q, Sun Y, Shi Z, He S (2017) Narrowband internet of things: Implementations and applications. *IEEE Internet Things J* 4(6):2309–2314
33. Liu Y, Liu A, Zhang N, Liu X, Ma M, Hu Y (2019) DDC: Dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks. *J Netw Comput Appl* 131:16–27
34. Li X, Liu A, Xie M, Xiong N, Zeng Z, Cai Z (2018) Adaptive aggregation routing to reduce delay for multi-layer wireless sensor networks. *Sensors* 18(4):751. <https://doi.org/10.3390/s18030751>.
35. Li Z, Xiao F, Wang S, Pei T, Li J (2018) Achievable rate maximization for cognitive hybrid satellite-terrestrial networks with AF-relays. *IEEE J Sel Area Comm Special issue on Advances in Satellite Communications* 26:304–313
36. Liu A, Zhao S (2018) High performance target tracking scheme with low prediction precision requirement in WSNs. *Int J Ad Hoc Ubiqu Co* 29:270–289
37. Peng X, Ren J, She L, Zhang D, Li J, Zhang Y (2018) BOAT: A block-streaming app execution scheme for lightweight IoT devices. *IEEE Internet Things J* 5(3):1816–1829
38. Chen X, Shi Q, Yang L, Xu J (2018) Thriftyedge: Resource-efficient edge computing for intelligent IoT applications. *IEEE Netw* 32(1):61–65
39. Liu Y, Liu A, Chen Z (2015) Analysis and improvement of send-and-wait automatic repeat-request protocols for wireless sensor networks. *Wirel Pers Commun* 81(3):923–959
40. Luo M, Wang K, Cai Z, Liu A, Li Y, Cheang CF (2019) Using imbalanced triangle synthetic data for machine learning anomaly detection. *CMC-Comput Mater Con* 58(1):15–26
41. Ren J, Zhang Y, Zhang N, Zhang D, Shen X (2016) Dynamic Channel access to improve energy efficiency in cognitive radio sensor networks. *IEEE Trans Wirel Commun* 15(5):3143–3156
42. Huang M, Liu A, Zhao M, Wang T (2018) Multi working sets alternate covering scheme for continuous partial coverage in WSNs. *Peer Peer Netw Appl.* 12(3):553–567. <https://doi.org/10.1007/s12083-018-0647-z>
43. Zhou C, Gu Y, Fan X, Shi Z, Mao G, Zhang Y (2018) Direction-of-arrival estimation for coprime array via virtual array interpolation. *IEEE Trans Signal Process.* <https://doi.org/10.1109/TSP.2018.2872012>
44. Pu L, Chen X, Xu J et al (2016) D2D fogging: An energy-efficient and incentive-aware task offloading framework via networks-assisted D2D collaboration. *IEEE J Sel Area Comm* 34(12):3887–3901
45. Wang T, Zhou J, Liu A, Bhuiyan M, Wang G, Jia W (2018) Fog-based computing and storage offloading for data synchronization in IoT. *IEEE Internet Things J.* 6(3):4272–4282. <https://doi.org/10.1109/JIOT.2018.2875915>
46. Huang CM, Chen YF, Xu S et al (2018) The vehicular social network (VSN)-based sharing of downloaded geo data using the credit-based clustering scheme. *IEEE Access* 6: 58254–58271
47. Liu Y, Ma M, Liu X, Xiong N, Liu A, Zhu Y (2018) Design and analysis of probing route to defense sink-hole attacks for internet of things security. *IEEE T Netw Serv Sci Eng.* <https://doi.org/10.1109/TNSE.2018.2881152>.
48. Li Z, Liu Y, Ma M, Liu A, Zhang X, Luo G (2018) MSDG: A novel green data gathering scheme for wireless sensor networks. *Comput Netw* 142(4):223–239
49. Liu W, Zhuang P, Liang H, Peng J, Huang Z (2018) Distributed economic dispatch in microgrids based on cooperative reinforcement learning. *IEEE T Neur Net Lear* 29(6):2192–2203
50. Ren J, Zhang Y, Zhang K, Liu A, Chen J, Shen X (2016) Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks. *IEEE T Ind Inform* 12(2):788–800
51. Ren Y, Liu Y, Zhang N, Liu A, Xiong N, Cai Z (2018) Minimum-cost Mobile crowdsourcing with QoS guarantee using matrix completion technique. *Pervasive Mob Comput* 49:23–44
52. Gui J, Hui L, Xiong N (2018) Enhancing cellular coverage quality by virtual access point and wireless power transfer. *Wirel Commun Mob Comput* 2018:9218239. <https://doi.org/10.1155/2018/9218239>
53. Luo X, Lv Y, Zhou M et al (2016) A laguerre neural networks-based ADP learning scheme with its application to tracking control in the internet of things. *Pers Ubiquit Comput* 20(3):361–372
54. Zhang H, Meng W, Qi J, Wang X, Zheng W (2019) Distributed load sharing under false data injection attack in an inverter-based microgrid. *IEEE Trans Ind Electron* 66(2):1543–1551
55. Byun H, Son S, So J (2013) Queue management based duty cycle control for end-to-end delay guarantees in wireless sensor networks. *Wirel Netw* 19(6):1349–1360
56. Lee JH (2013) A traffic-aware energy efficient scheme for WSN employing an adaptable wakeup period. *Wirel Pers Commun* 71(3): 1879–1914
57. Xie R, Liu A, Gao J (2016) A residual energy aware schedule scheme for WSNs employing adjustable awake/sleep duty cycle. *Wirel Pers Commun* 90(4):1859–1887
58. Teng H, Zhang K, Dong M, Ota K, Liu A, Zhao M, Wang T (2018) Adaptive transmission range based topology control scheme for fast and reliable data collection. *Wirel Commun Mob Comput* 2018: 4172049. <https://doi.org/10.1155/2018/4172049>
59. Liu A, Zheng Z, Zhang C, Chen Z, Shen X (2012) Secure and energy-efficient disjoint multipath routing for WSNs. *IEEE Trans Veh Technol* 61(7):3255–3265
60. Zhang J, Liu A, Hu P, Long J (2018) A fuzzy-rule-based packet reproduction routing for sensor networks. *Int J Distrib Sens N* 14(4). <https://doi.org/10.1177/1550147718774016>
61. Dam TV, Langendoen K (2003) An adaptive energy-efficient MAC protocol for wireless sensor networks. *Proceedings of the 1st international conference on Embedded networked sensor systems.* ACM 171–180
62. Huang M, Liu W, Wang T, Song H, Li X, Liu A (2019) A queuing delay utilization scheme for on-path service aggregation in services oriented computing networks. *IEEE Access.* 7(1):23816–23833. <https://doi.org/10.1109/ACCESS.2019.2899402>



Qiaoyan Li is currently a student at the School of Computer Science and Engineering, Central South University, China. Her research interests include networked cyber-physical systems and wireless sensor networks. E-mail: qiaoyanli@csu.edu.cn.



Mande Xie was born in 1977 and is currently a Professor in the Zhejiang Gongshang University. He received the PhD degree in Circuit & System from Zhejiang University in 2006. His research interests include Wireless Sensor Networks (WSNs), Social Network and Privacy Preservation. E-mail: xiemd@zjgsuedu.cn.

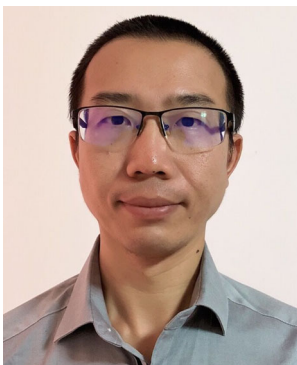


Anfeng Liu is a Professor in the School of Computer Science and Engineering of Central South University, China. He is also a Member (E200012141M) of China Computer Federation (CCF). He received the M.Sc. and Ph.D. degrees from Central South University, China, 2002 and 2005, both in computer science. His major research interests are crowd sensing network and wireless sensor network. E-mail: afengliu@mail.csu.edu.cn.



Neal N. Xiong is current an Associate Professor at Department of Mathematics and Computer Science, Northeastern State University, OK, USA. He received his both PhD degrees in Wuhan University (about sensor system engineering), and Japan Advanced Institute of Science and Technology, respectively. Before he attends Northeastern State University, he worked in Georgia State University, Wentworth Technology Institution, and Colorado Technical University

about 10 years. His research interests include Cloud Computing, Security and Dependability, Parallel and Distributed Computing, Networks, and Optimization Theory. Dr. Xiong published over 200 international journal papers and over 100 international conference papers. He is serving as an Editor-in-Chief, Associate editor or Editor member for over 10 international journals, E-mail: dnxiong@126.com.



Tian Wang received his BSc and MSc degrees in Computer Science from the Central South University in 2004 and 2007, respectively. He received his Ph.D. degree from the City University of Hong Kong in 2011. Currently, he is an associate professor at the National Huaqiao University of China. His research interests include wireless sensor networks, social networks and mobile computing. E-mail: wangtian@hqu.edu.cn.