

RESEARCH

Open Access



# Learning-based synchronous approach from forwarding nodes to reduce the delay for Industrial Internet of Things

Minrui Wu<sup>1</sup>, Yanhui Wu<sup>2</sup>, Xiao Liu<sup>1</sup>, Ming Ma<sup>3</sup>, Anfeng Liu<sup>1\*</sup>  and Ming Zhao<sup>4</sup>

## Abstract

The Industrial Internet of Things (IIoTs) is creating a new world which incorporates machine learning, sensor data, and machine-to-machine (M2M) communications. In IIoTs, the length of the transmission delay is one of the pivotal performance because dilatory communication will cause heavy losses to industrial applications. In this paper, a learning-based synchronous (LS) approach from forwarding nodes is proposed to reduce the delay for IIoTs. In an asynchronous Media Access Control protocol, when senders need to send data, they always require to wait for their corresponding receiver to wake up. Thus, the delay here is greater than in the synchronous network. However, the synchronization cost of the whole network is enormous, and it is difficult to maintain. Therefore, LS mechanism uses a partial synchronization approach to reduce synchronization costs while effectively reducing delay. In LS approach, instead of synchronizing the nodes in the entire network, only sender nodes and part of the nodes in their forwarding node set are synchronized by self-learning methods, and accurate synchronization is not required here. Thus, the delay can be effectively reduced under the low cost. Secondly, the nodes near sink maintain the original duty cycle, while the nodes in the regions away from the sink use their remaining energy and perform synchronization operations, so as not to damage the network lifetime. Finally, because the synchronization in this paper is based on different synchronization periods among different nodes, it can improve the network performance by reducing the conflict between simultaneous data transmission. The theoretical analysis results show that compared with the previous approach FFSC, LS approach can reduce the end-to-end delay by 5.13–11.64% and increase the energy efficiency by 14.29–17.53% under the same lifetime with a more balanced energy utilization.

**Keywords:** Industrial Internet of Things, Learning-based synchronous (LS) approach, Communication delay, Duty cycle, Network lifetime

## 1 Introduction

The Internet of Things (IoT) has been regarded as the third wave of information technology [1–12]. Based on IoT technologies, the Industrial Internet of Things (IIoTs) [13] incorporates machine learning and big data technology, sensor data, and machine-to-machine (M2M) communications that have existed in industrial areas for years [14–16]. In IIoTs, huge amounts of wireless sensor nodes [17–24] can be deployed more conveniently than the previous wired network to monitor

the whole process of industrial production from many angles, in real time or semi-real time. In such way, IIoTs is creating a new world for industrial manufacturing, where the workers or managers can manage their industrial manufacturing in more informed ways and can make more opportune and better informed decisions. Many advanced IIoTs have been successfully applied in industrial manufacturing. Wireless sensor-based networks [24–27] are one of the important components for IIoTs, in which each node sensing manufactures information from surrounding environment and each node in the network helps in routing to data center (DC) or sink by forwarding data of other nodes [27–30]. One of the most important and challenging issue is delay [4, 6, 24,

\* Correspondence: afengliu@mail.csu.edu.cn

<sup>1</sup>School of Information Science and Engineering, Central South University, 401 room, Computer building, Changsha 410083, China  
Full list of author information is available at the end of the article

29, 31]; especially, data end-to-end delay is very important for industrial manufacturing [32], because there are often some emergency situations that require workers to take corresponding emergency measures to handle. Therefore, it is important to quickly send the perceived information to the sink. So it is very important to study how to reduce the delay effectively for IIoTs, edge network, and society network as well as cloud computing [33–35], and there are also some related research work [4, 6, 24, 29, 32, 36–38].

In wireless sensor-based network, the data of nodes is usually routed to the sink via multi-hop relay. When a sender has data to send, they generally chose nodes which are within its own sending range and closer to its own distance from sink as forwarding nodes (FNs) [4]. FNs represent the sets of the candidate forwarding nodes of the senders. In asynchronous-based wireless sensor-based network, each node adopts a periodic awake/sleep mode to save energy, because the consumption of a node in the awake state is 100 to 1000 times as much as in the sleep state [4]. This method is called duty cycling. In order to save energy consumption, nodes should be left in the sleep state as much as possible. But when nodes in the sleep state, they cannot receive or send data, which will increase the delay of data transmission. When senders have data that needs to be sent to sink, their forwarding nodes may be in the sleep state. Therefore, senders need to wait for their FNs to awake before they can transmit data. In general, due to their inherent characteristics of constrained resource for sensor nodes, the duty cycle of each node is relatively small and its sleeping time is longer than waking time in order to save energy. This means that the time that nodes in the sleep state is longer than in the awake state, while the time that nodes in the awake state is generally longer than the time it takes to transmit a packet effectively, so that a packet can complete the transmission in a cycle. Obviously, when the sender has only one forwarding node, before it transmits data, the expected length of time required to wait for the receiver to wake up (called sleep delay) is half of the length of its sleeping time. Thus, in the network with a duty cycle of 0.2, the required time that the sender waits for the receiver to wake up is twice as much as that it takes to transmit data packets. Therefore, sleep delay is an important component of end-to-end delay and plays an important role in it. In the data transmission, the time required to send data packets is needed in any network and cannot be reduced; thus, to reduce the end-to-end delay of packets, the most critical thing is to reduce the sleep delay per hop transmission. The most direct way to reduce the sleep delay is to increase the duty cycle of nodes. If the duty cycle of the node is increased to 1, then the sleep delay is reduced to 0. At present, there are some research to reduce the

delay by adjusting the duty cycle of nodes [4, 32]. But in general, this method of increasing duty cycle of nodes is difficult to be applied in practice because it greatly increases the energy consumption of nodes. Another way to reduce sleep delay is by synchronizing the network with nodes waking up and sleeping at the same time, so that nodes can communicate with each other when they are awake [4, 32]. Although there is no sleep delay, the disadvantage of this approach is that each node in the network needs to be synchronized, while maintaining synchronization in large-scale networks requires a great deal of communication overhead. Moreover, as time goes on, the clock between nodes will produce an offset, so it is necessary to constantly maintain synchronization, and its cost is also very large. It can be seen that in the current research, to reduce the delay of the network will generally reduce the network life. In view of the deficiencies in previous research, a learning-based synchronous (LS) approach from forwarding nodes is proposed to reduce the delay for IIoTs while not reducing the network lifetime. The main contributions of this paper are as follows:

- (1). A LS approach from forwarding nodes is proposed to reduce the delay for IIoTs. In LS approach, for each sender, a part of nodes called candidate relays are selected from forwarding nodes which are closer to the sink than the unselected forwarding nodes. The sender synchronizes its duty cycle with its candidate relays by self-learning. Therefore, when the sender is awake, its candidate relays are also awake, thus greatly reducing the sleep delay. In addition, the data forwarding of senders are all carried out through their corresponding candidate relays, so the distance of each hop on the path from the sender to the sink is longer than the previous strategies that select awake nodes from all forwarding nodes. Therefore, the number of hops can be reduced, and the end-to-end delay can also be reduced. Moreover, in LS approach, it does not require that all nodes in the network remain synchronized, and only the synchronization between senders and their candidate relays is required. There is no strict requirement for synchronization here, so it is easy to implement.
- (2). In LS approach, the network lifetime is longer. In sensor-based network, the energy consumption of the nodes near the sink is high, while the energy consumption of the nodes far away from the sink is low and there have residual energy. Therefore, in the LS approach, in the range of one hop near the sink, the operations of the nodes are exactly the same as the previous strategy, but nodes whose distance to the sink are longer than the length of one

hop start self-learning to synchronize with their candidate relays, so they can just use their residual energy to keep the synchronization with their candidate relays. Thus, this approach can effectively utilize the network energy, while not reducing the network lifetime.

- (3). The effectiveness of LS approach is evaluated through extensive theoretical analyses, and we demonstrate that compared with the previous approach, LS approach has two critical advantages: (a) it significantly reduces the end-to-end delay of data transmission, and specifically, under the same lifetime, compared to previous methods, LS approach can reduce end-to-end delay by about 5.13 to 11.64% and (b) it takes full advantage of the residual energy to maintain the synchronization, and the network energy utilization ratio can be enhanced by as much as 15.91% averagely.

The rest of this paper is organized as follows. In Section 2, the related works are reviewed. The system model and problem statements are described in Section 3. Section 4 elaborates on the design of learning-based synchronous (LS) approach from forwarding nodes for IIoTs. The performance analysis of LS approach is provided in Section 5. Section 6 provides the simulation result. Finally, we conclude in Section 7.

## 2 Related work

A large number of sensor nodes are usually deployed in industrial manufacturing, forming the so-called Industrial Internet of Things (IIoTs). In such kind of application scenarios, sensory data are valid only when they reach the sink within the specified time threshold. In particular, alarm information obtained by sensor nodes need to be delivered as soon as possible; otherwise, it will cause great damage and disaster to industrial manufacturing [4, 6, 24, 29, 32, 36, 38–41]. Therefore, there are lots of research trying to reduce delay [2]. This paper divides the work into the following categories:

- (1). Routing-based method for reducing delay. Because of the limited communication capacity and communication range of sensor nodes, this method generally uses multi-hop routing to send data to the sink via several relays. But in general, a node has at least one or more forwarding nodes closer to the sink than itself. Thus, selecting the right forwarding nodes can effectively reduce delay. If the forwarding node which has the nearest distance to the sink is selected as relay node each time, the one-hop distance in the direction that is sent to the sink will be maximized, so that the number of hops on the path of transmission is minimal. Because each data

forwarding will increase the forwarding delay, reducing the number of forwarding can effectively reduce the delay. This method is the first proposed shortest routing method. Based on the shortest routing method, an optimization method for multiple integrated routing is proposed. The general characteristic of such routing methods is as follows: the selection of relay node is based on the composite values of multiple performance indexes, and the forwarding nodes with high integrated value will be selected as relay nodes. If the requirement for delay is high, the weight of delay will be large; for example, when only the delay is considered, the shortest routing method is appropriate. If the energy consumption and delay are considered synthetically, the forwarding nodes with high residual energy and closer to the sink are selected as relay nodes, so that the energy consumption is balanced and the delay is smaller. Such methods generally assume that nodes are always in the awake state; therefore, as long as a sender has data to transmit, its forwarding nodes are optional. But in fact, most sensor networks adopt awake/sleep periodic rotation to save energy. Thus, such methods do not apply to sleep-wake cycling sensor networks.

For sleep-wake cycling sensor networks, Naveen and Kumar [42] proposed an approach for relay node selection for sleep-wake cycling wireless sensor networks. Their goal is how to select the appropriate relay node from the forwarding nodes to minimize the delay. The approach proposed by Naveen and Kumar takes into account the fact that when the sender selects the relay node, the forwarding node which is closer to the sink is in the sleep state. At this point, the sender has two choices: one is to wait for the node which is closer to the sink to wake up, and the other is to select the forwarding node that is awake but not near the sink as the relay node. If the sender waits for a node near the sink to wake up, although this method can reduce the number of hops required for routing, the delay of this hop will increase as a result of waiting. If the node which is in the awake state now is selected as the relay node, the number of hops will increase, so it is possible that the delay is not necessarily small. Naveen and Kumar [42] consider that the selection of relay nodes can be inducted as asset selling problems. The description is that the seller wants to sell his product, the buyer arrives at any time, and the price is stochastic. If the seller does not buy a deal with the buyer, the buyer will leave and never come back. The new arrival buyer may have a lower price than the previous buyer. The selection of relay node is similar to that. If a forwarding node is awake, the sender should make a decision whether to

select it as relay node. If the sender decides to continue waiting, the new node which is in the awake state may be farther away from the sink than the previous nodes. Naveen and Kumar convert the distance from the sink to the forwarding node and the waiting time to reward, then select the node with high reward as relay node.

A Fast data collection for the node Faraway sink and Slow data collection for the node Close to sink (FFSC) approach is proposed to meet energy efficient, and the end-to-end delay can be minimized in Ref. [4]. In FFSC approach, for the region which is far away from the sink, nodes adopt a routing strategy of forwarding immediately if there is a forwarding node available. Thus, the remaining energy of the node in this region can be fully utilized. For nodes closer to the sink, the optimized relay nodes need to be selected before forwarding so as to save energy and maintain high network lifetime. Therefore, as a whole, the energy efficient utilization of FFSC approach is improved, and the network delay is reduced [4].

The advantages of routing-based method for optimizing delay are its better versatility and a wide range of applications. The deficiency is that the design method is more complex, and the performance is limited. The main reason for the delay in such methods is that nodes are not synchronized, so forwarding nodes are not necessarily awake when the sender has data to transmit, which result in longer delay.

- (2). The solution based on duty cycle. As can be seen from the previous discussion, if the nodes are always in the awake state, there is no sleep delay when the sender has data to transmit. But because the energy consumption of nodes when they are in the awake state is high, it is not an effective way to simply increase the duty cycle. Chen et al. [32] proposed a method of changing duty cycle to reduce delay. They found that the energy consumption of nodes in the region closer to the sink is high, while the energy consumption of nodes in the region far away from the sink is low, so there is a great deal of residual energy. Based on the discovery, the nodes in the region far away from the sink adopt large duty cycle to reduce delay, and the nodes in the region closer to the sink adopt small duty cycle in order to ensure the longer network lifetime, thus reducing the overall delay and maintaining a high lifetime.

Lee [43] also proposed an approach to adjust the duty cycle of nodes according to the traffic of nodes, which is called Adaptable Wakeup Period (AWP) [43]. They find many performances in the network are related to duty cycle. For example, if the duty cycle increases, the

successful delivery ratio becomes high and retransmission counts decrease. Based on the above discovery, in order to save energy, when the traffic of the node is small, the duty cycle of this node should be reduced, so that the energy can be saved without reducing the performance of the network. When the traffic of the node is large, the duty cycle of this node is increased to meet the performance requirements of the network. Thus, compared with the approach that the whole network uses the same duty cycle (SDC), as a whole, the AWP approach reduces the energy consumption of the network without reducing network performance.

- (3). The approach based on synchronous network [44]. The description above is mainly about asynchronous network. Another type of network is synchronous network, in which each node follows the same clock frequency, so that the nodes wake up only at the time they take to work, while sleeping at other times. In such methods, nodes only need to be awake at time slots for work, thus saving energy effectively. However, such methods require precise synchronization, which limits the scalability of the system. Because the synchronization in a large-scale network is a very difficult task, in the wireless sensor networks, most applications still adopt asynchronous operation. Thus, each node only needs to select its own working slots independently without synchronization, which can increase its applicability. But in general, when using asynchronous operation mode, the performance of wireless sensor networks is not as good as synchronous network.
- (4). Communication method based on competitiveness. TDMA-based Media Access Control (MAC) protocol is a non-competitive communication protocol [23]. In such a protocol, the system assigns the appropriate work slot to each node, and each node wakes up in the planned slot, performs the corresponding data operation, and then switches to sleep. This is a very energy-saving and efficient way. In this approach, delay usually refers to the number of time slots needed to collect a round of data in the network. In Ref. [45], the authors came up with centralized scheduling algorithm, in which the radius of network is  $R$ , the maximum node degree is  $\Delta$ , and the bound of delay is  $23 + R\Delta + 18$  time slots. The distributed scheduling method proposed by Yu et al. [46] can generate conflict-free schedules where the upper limit of delay is  $24D + 6\Delta + 16$  time slots, and the diameter of network is  $D$ . Xu et al. [47] gave theoretical proof that their algorithm can generate the aggregation schedule where the upper limit of delay is  $16 + R\Delta - 14$  time slots. This approach is often used in wireless sensor nodes for data fusion,



especially in the network where  $n$  packets are fused into one packet. But the deficiency of this approach is that the operation of nodes in the network should be determined in advance. Once the node time slot is determined, the entire network must adopt planned sequence of operations and cannot change. Therefore, this approach cannot be applied to networks whose data operations cannot be determined in advance. In industrial applications, sensor nodes often needs to be aware of unexpected events, so it is impossible to determine its work slot in advance. Thus, in most cases, this method cannot be adopted.

In conclusion, the main reason for the delay is the asynchronous operation of the node, which causes the forwarding nodes to be in the sleep state when the sender has data to transmit. Although the delay will be effectively reduced when using synchronous network, the maintenance cost of synchronous network has also reduced the development of its application. Therefore, this paper proposes a scheme of partial synchronization, which reduces the cost and operation difficulty of synchronization effectively and reduces the delay effectively. At the same time, partial synchronization between the sender and its forwarding set is the difference between this and other similar scheme.

### 3 The system model and problem statement

#### 3.1 The network model

In this paper, the system model is based on duty-cycled wireless sensor networks [4, 32]. This kind of network adopts a common method in WSNs, duty cycling, in which nodes transform between active and sleep modes. A wakeup interval is always predefined there, which is the basis of the transformation. Combined with the opportunistic routing based on asynchronous protocol, nodes wake up at different moments and each node has multiple forwarders. Each node has a communication range; here, we think of it as a circular region with a radius of  $r$  and make the sender as the center of the circle. Other nodes in this circle have the chance to become the forwarding nodes of the sender. When a sender wants to transmit a packet, at least one of its corresponding candidate forwarders should be in an active mode. Otherwise, the sender has to wait for the receiver to wake up. In the process of being transmitted to the sink node, a packet may be forwarded by multiple relay nodes, which is affected by the metric for forwarder selection.

Due to the random and uniform distribution of the monitoring targets in the entire network, the probabilities that a target being perceived by any node are equal. Similarly, for each node, the probability of generating data is equal. The security of network such as privacy and attacks are not considered in this paper [48, 49].

#### 3.2 System parameters

As is described above, each node has a duty cycle, waking up and sleeping periodically. Any node cannot receive or send packets while it is in sleep mode. Beyond that, in the sleep state, less energy is consumed compared to the active state. In addition to transmission, each sender does periodic self-learning. After self-learning, the sender starts communication. Thus, the duration of communication consists of two main parts: active periods and sleep periods. In a self-learning duty cycle, nodes always stay active and detect the state of the corresponding forwarding nodes. According to the description above, we use the following equations to denote the nodes' duty cycle, which is denoted as  $C$  below:

$$C_{COM} = (D_{LEA} + D_{COM}^A) / (D_{LEA} + D_{COM}^A + D_{COM}^S) \tag{1}$$

$$C_{LEA} = D_{LEA}^A / D_{LEA} = D_{LEA}^A / D_{LEA}^A \tag{2}$$

where  $D_{COM}$  is the duration of communication and  $D_{COM}^A$  and  $D_{COM}^S$  are the periods of the nodes when they are in active and sleep modes respectively.  $D_{LEA}$  is the learning duration of the nodes, and because at this time nodes are always active, here,  $D_{LEA}$  is equal to  $D_{LEA}^A$ .

The calculation of power consumption mainly include these three aspects: the power consumed in learning duration, the power consumed in receiving or transmission,

**Table 1** System parameters

Parameter	Description
$E_O$	Initial energy
$D_{LEA}$	Self-learning duration
$D_{COM}$	Communication duration
$\epsilon_T$	Power consumption of transmission
$\epsilon_R$	Power consumption of reception
$\epsilon_{LEA}$	Power consumption of self-learning
$\epsilon_S$	Power consumption of sleeping
$D_B$	Preamble duration
$D_{ACK}$	Acknowledge window duration
$D_P$	Packet duration
$C_{COM}$	Communication duty cycle
$C_{LEA}$	Self-learning duty cycle
$E_{LPL}$	Energy consumption in low power listening
$E_T$	Energy consumption in transmission
$E_{LEA}$	Energy consumption in self-learning
$E_R$	Energy consumption in receiving
$\gamma$	Network delay
$E_U$	Energy utilization
$r$	The transmission range of a node
$\rho$	Node density
$T_N$	Network lifetime

and the power required when nodes in the sleep mode. Power consumption is connected with duty cycle.

The main system parameters are listed in Table 1 [4, 42]. Other parameters will be explained when they appear in equations.

### 3.3 Problem statement

Each packet is sent to the sink after being forwarded by multiple relay nodes. In regard to the packet transmission, the main objective of this work is to propose a method for minimizing delay and maximizing network lifetime. These indicators are explained below.

(1)The effective energy utilization

There are  $n$  nodes in the network.  $E_U$  is the energy utilization, which relates to the ratio of the energy utilized by the whole network and the total initial energy in the network. The energy utilization can be expressed as the equation below. The maximization of energy utilization is to improve the efficient utilization of network energy and maximize the ratio of energy consumption to the initial energy in the network.

$$\max(E_U) = \max \left[ \left( \frac{\sum_{1 \leq i \leq n} \varepsilon_i}{\sum_{1 \leq i \leq n} E_O^i} \right) \right] \quad (3)$$

As shown in the equation,  $\varepsilon_i$  is the energy consumption of the  $i$ th node and  $E_O^i$  represents the initial energy of node  $i$ .

(2)The maximization of network lifetime

Network lifetime is defined as the period from the beginning until the first node dies out because of energy drain. Thus, maximizing the lifetime of the first node which uses up its energy can make the network lifetime maximized. Here, the initial energy of node  $i$  is  $E_O^i$ , and the energy consumption of node  $i$  is  $\varepsilon_i$  per unit time.

Accordingly, this requirement is guaranteed in the following equation.

$$\max(T_N) = \max \left( \min_{1 \leq i \leq n} \left( \frac{E_O^i}{\varepsilon_i} \right) \right) \quad (4)$$

(3)The minimization of end-to-end delay

The end-to-end transmission delay refers to the time that the nodes take for transmitting the packet from the sender to the sink. Assuming that in the packet transmission, the packet passes through  $k$  nodes. Here, let  $Y_{E2E}$  represents the delay and  $y_{i-1, i}$  represents the delay of the transmission from  $(i-1)$ th to  $i$ th node. Then, the corresponding equation can be obtained.

$$\begin{aligned} \text{Min}(Y_{E2E}^{v_i}) &= \min \left( \sum y_{v_j, v_k} \right) \\ &| \forall v_j, v_k \in P_{v_i} \text{ and } v_k \text{ is forwarding node of } v_j \end{aligned} \quad (5)$$

where  $P_{v_i}$  represents the routing path from the node  $v_i$  to the sink and  $v_j \in P_{v_i}$  expresses that  $v_j$  is a node in the  $P_{v_i}$ .

In conclusion, the objectives of our research are

$$\begin{cases} \max(E_U) = \max \left[ \left( \frac{\sum_{1 \leq i \leq n} \varepsilon_i}{\sum_{1 \leq i \leq n} E_O^i} \right) \right] \\ \max(T_N) = \max \left( \min_{1 \leq i \leq n} \left( \frac{E_O^i}{\varepsilon_i} \right) \right) \\ \text{Min}(Y_{E2E}^{v_i}) = \min \left( \sum y_{v_j, v_k} \right) \end{cases} \quad (6)$$

## 4 Design of LS approach

### 4.1 Research motivation

Opportunistic Routing in Wireless sensor network (ORW) is a routing protocol and works on top of an asynchronous MAC protocol [50]. In this protocol, each

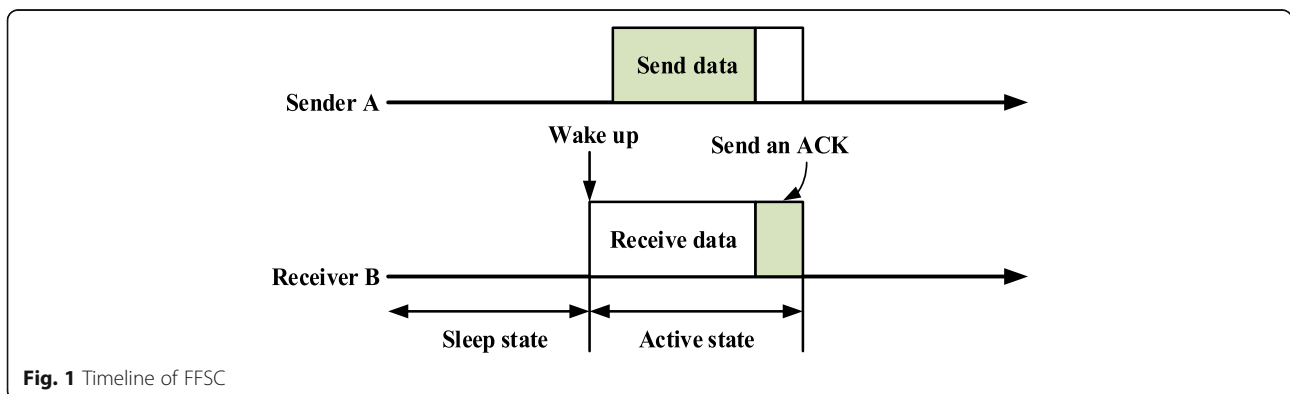


Fig. 1 Timeline of FFSC

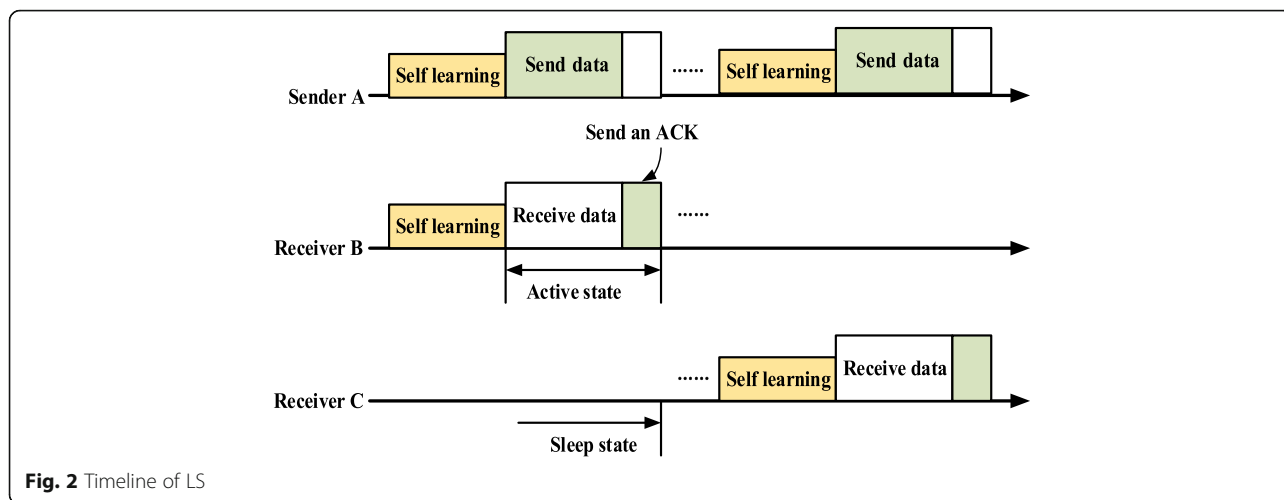


Fig. 2 Timeline of LS

sender has multiple candidate forwarding nodes and each forwarding nodes has the chance to receive the packet when the sender has data to transmit. When the first forwarding node receives the packet and sends the acknowledgement (ACK) to sender, the sender can finish its transmission. ORW uses a metric called Expected Duty Cycled Wakeups (EDC) to select relay nodes. EDC is a measure of the time from the sender transmitting a packet to the packet reaching the sink, which does not consider the location of relays. Thus, there is a probability that a node far away from the sink is selected as forwarder, which may cause longer packet delay. And in the FFSC approach, nodes closer to the sink should select the optimized relay nodes before forwarding, while nodes far away from the sink adopt the strategy of forwarding immediately if there is a forwarding node available. However, because nodes are not synchronized, the sender wait time will be longer and lead to the longer delay. Therefore, we are committed to designing a better way to reduce delay.

- (1). Synchronization can reduce delay. As can be seen in Fig. 1, without synchronization, the expected delay is half of sleep time in the FFSC approach. If the duty cycle of the corresponding nodes is roughly synchronized before sending the data, the sender does not require to wait for the receiver to wake up or only need to wait for a shorter time than without synchronization. As Fig. 2 shows, after self-learning, the synchronization of the sender and its receiver is completed, so when sender A transmits data, one of its candidate receiver is also in the awake state, and then, the delay is reduced.
- (2). It is difficult to implement synchronization. The trouble is that maintaining a wide range of

synchronization consumes a lot of energy. But the approach presented in this paper does not require synchronization of all nodes in the entire network and only requires synchronization within the forwarding range of the sender, so it is easy to implement. Furthermore, nodes which are closer to the sink do not require to maintain synchronization, so these nodes do not need additional energy consumption, while nodes which are far away from the sink have residual energy. Using this part of energy for synchronization, the delay can be reduced without affecting the network lifetime.

- (3). Duplicate packets will lead to the waste of energy consumption. But this problem is inevitable while reducing delay as much as possible. Because multiple forwarding nodes may be awake at the same time, duplicate packets will be generated whether forwarding nodes transmit the same packet to their next nodes or return the ACK to

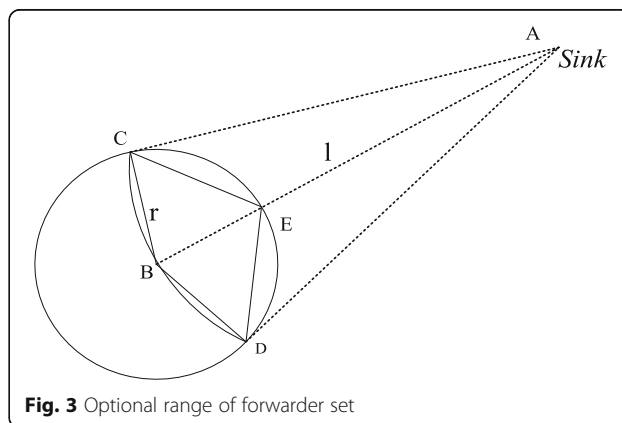


Fig. 3 Optional range of forwarder set

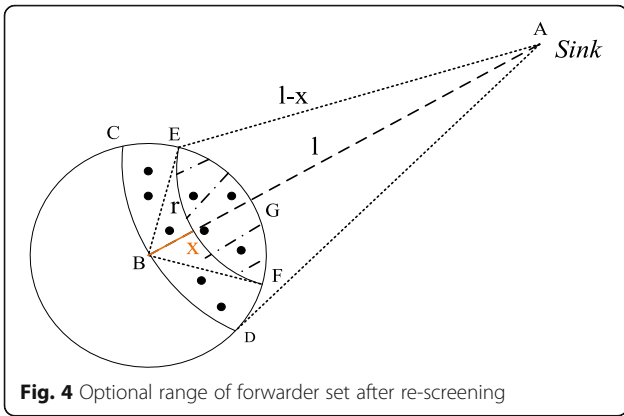


Fig. 4 Optional range of forwarder set after re-screening

the sender. In the approach proposed in this paper, the selection of nodes is optimized, and for each sender, only its forwarding nodes within a small range will be synchronized, in order to avoid aggravating the problem of duplicate packets compared to other approaches.

4.2 LS approach

In this paper, the forwarder set of each node is defined as a region formed by the intersection of a circle and an arc through the center of the circle. As shown in Fig. 3, the distance between sink and the sender node B is  $l$  and the radius of the circle is equal to the communication range of node B. In order to avoid selecting forwarders with longer distance from the sink, there is a constraint that the distance from the candidate relay node to the sink is at least closer than node B to sink.

The area of this region can be calculated by equation:

$$\vartheta_0 = l^2 \cdot \cos^{-1}\left(1 - \frac{r^2}{2l^2}\right) + r^2 \cdot \cos^{-1}\frac{r^2}{2l} - l \cdot r \cdot \sin\left(\cos^{-1}\frac{r}{2l}\right) \tag{7}$$

where the first part of Eq. (7) is used to calculate the area of sector ACD, the second part represents sector BCD area, and the third calculates the area of the quadrilateral ACBD.

Based on this model, the selection of relay nodes and the determination of duty cycle are related to self-learning. First, we define the nodes in the shaded portion of Fig. 4 as the candidate relays.

Assuming that the density of relays is  $\rho$ . The area of the shaded portion can be expressed as

$$\begin{aligned} \vartheta = \vartheta_{fanAEF} + \vartheta_{fanBEF} - \vartheta_{\blacksquare AEBF} = & \left( \cos^{-1}\frac{(l-x)^2 + l^2 - r^2}{2(l-x)l} \right) \\ & \cdot (l-x)^2 + \left( \cos^{-1}\frac{l^2 + r^2 - (l-x)^2}{2lr} \right) \\ & \cdot r^2 - l \cdot r \cdot \sin\left(\cos^{-1}\frac{l^2 + r^2 - (l-x)^2}{2lr}\right) \end{aligned} \tag{8}$$

Thus, the number of forwarders in forwarder set of node B (denoted as  $\sigma$ ) is

$$\sigma = \rho \cdot \vartheta \tag{9}$$

The sender node stays active in the self-learning cycle and sends preamble repeatedly to detect the state of its candidate relays in the forwarder set. After sensing their duty cycle, we need to determine a period of time where

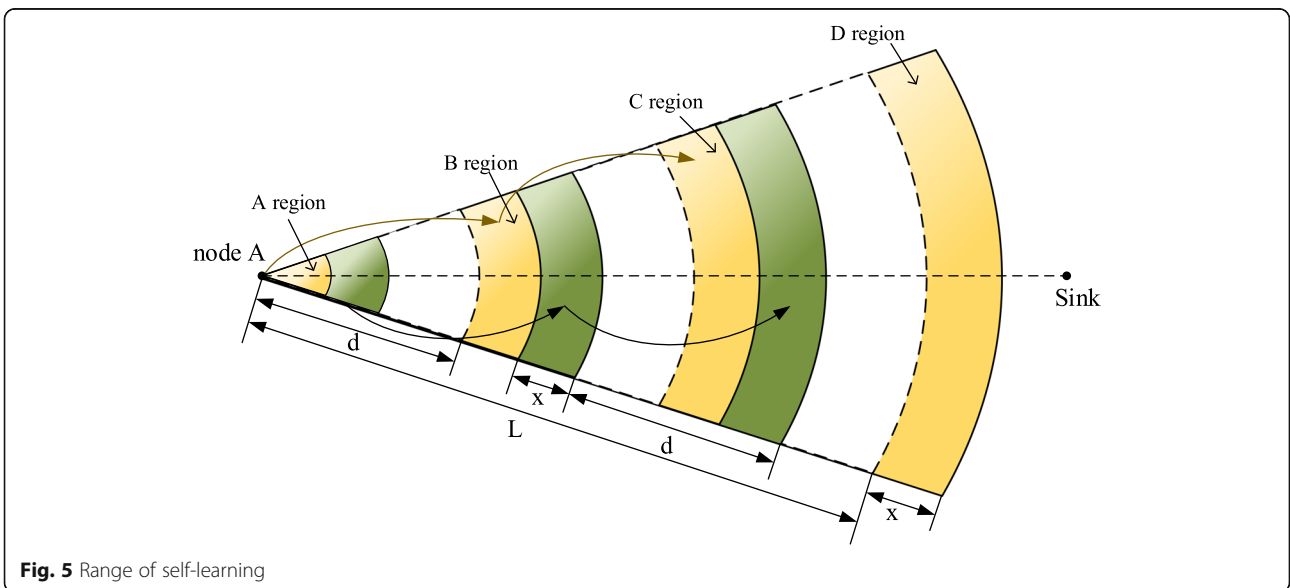


Fig. 5 Range of self-learning



more nodes are active. Compared with sender's duty cycle, if sender's active period is nearly coincident with the period where the majority of forwarders are active, there is no necessary to change sender's duty cycle temporarily. Beyond that, the distance between forwarders and sink also needs to be taken into account. In short, we should change the duty cycle of the sender, in order to synchronize the duty cycle of the nearest forwarder to sink and the sender. This approach can make one-hop distance longer and reduce delay. Meanwhile, each node remembers the number of its corresponding nodes participating in the adjustment of the cycle. After the sender has adjusted its duty cycle, the relay nodes affected by it also have to adjust their cycle. Another method is to keep the original start time of node's cycle constant and change the main duty cycle, which cannot affect the work of the related nodes. Certainly, this method will lead to the fragmentation of the cycle; therefore, adjustment and consolidation are also necessary at last.

In several self-learning cycles, the first task to complete is to divide the area from the sender to the slot into multiple regions with a width of  $x$ , as shown in Fig. 5. The distance between each of the two areas that need to be synchronized is  $d$ , synchronizing the duty cycle of the nodes near the sender node A with the nodes in the next region closer to the sink. Then, the nodes in this area repeat the above operations until the region containing the sink. Synchronization of any two regions is completed in this multiple self-learning cycle. But it is important to note that when the A region is learning from the B region, the B region is in the normal duty cycle, so the learning of the B region toward the C region should begin after the completion of the learning of the A region. However, there is another approach that when the A region is learning from the B region and the C region is learning from the D region, the C region can learn from the B region at the same time. But one drawback of this approach is that it increases the probability of ACK conflict.

At the same time, because there are more than one sender, the intersection of candidate forwarding node sets of different senders may not be a non-empty set. In this case, the solution we take is that for any two senders, if their intersection of forwarder sets is larger, we synchronize them with their forwarding nodes; if their intersection of forwarder sets is smaller, we divide the intersection into two parts and realize the synchronization of two senders respectively. In short, the number of forwarding nodes of any sender cannot be too small.

In a complete synchronization, a self-learning duration is  $D_{LEA}$  and the duty cycle of each relay node is  $c^A/c$ . In that way, assuming that a self-learning cycle can be divided into several time slots and the length of a time slot is  $c$ , then there are  $D_{LEA}/c$  time slots. The probability of any node waking

up in each slot is  $c^A/c$ . In the above context, each node has  $\sigma$  forwarders, and before a node sends a data packet, it sends the preamble at a certain frequency (denoted as  $V$ ) in the self-learning state. It is assumed that the average distance between any two nodes in the two regions is  $d$  and that a node should be at least  $2Vd$  after the delivery of a packet and then send the probe packet. When a node receives a probe packet, it sends an ACK back to the sender. In addition to ACK conflict problem, there is no guarantee that the ACKs received by the sender nodes are not returned by the nodes that returned ACK before. Therefore, it is a probability problem to estimate the time spent in self-learning.

The whole process of LS approach can be described in Algorithm 1.

---

**Algorithm 1:** LS approach
 

---

**Input:** The initial duty cycle and energy of nodes.

**Part I: The selection of forwarding nodes:**

- 1: **For** each sender node  $v_i$  **Do**
- 2:   In the transmission range of  $v_i$ , if the distance from the node  $v_j$  to sink is at least closer than  $v_i$  to the sink,  $v_j$  can be selected as candidate forwarding nodes
- 3: **End For**

**Part II: The self learning period:**

- 4: **For** each sender node  $v_i$  **Do**
- 5:   Select its forwarding nodes set
- 6:   The sender remains active and sends preamble repeatedly to detect the state of the nodes in the forwarder set while all forwarding nodes switch between sleep and active modes normally
- 7:   Adjusting the duty cycle of sender and its forwarding nodes to maximize their coincident activity time
- 8: **End For**

**Part III: Transmission:**

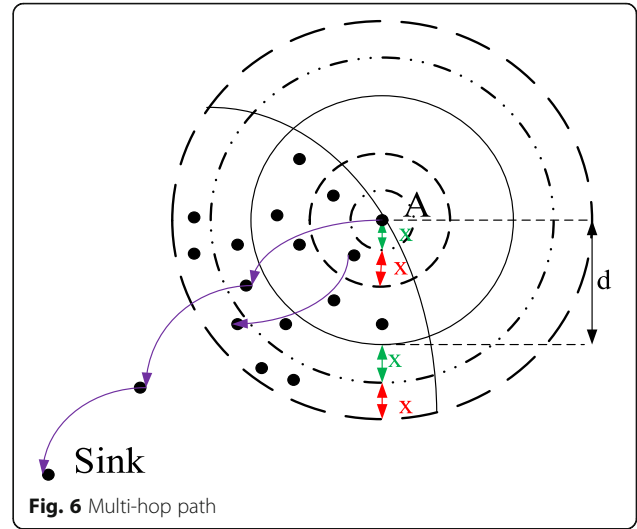
- 9: **For** each sender node  $v_i$  **Do**
  - 10:   **If** the sender has data to send **then**
  - 11:     **If** at least one node receives data and returns *ACK*
  - 12:       then
  - 13:         Successfully completed a single hop transmission
  - 14:       **End if**
  - 15:       **Else**
  - 16:         Wait for a while and continue sending;
  - 17:       **End else**
  - 18: **End for**
- 

Based on the description above, there are  $D_{LEA}/c$  time slots, where  $D_{LEA}$  is the duration of self-learning and  $c$  is the length of a time slot. When multiple nodes in the self-learning area are sending probe packets in the similar time, it is easy to cause

collisions on the sending and receiving channels. Thus, in addition to the time interval of each node that sends probe packets, it is necessary to restrict the interval at which each node sends a probe packet and its means is based on Carrier Sense Multiple Access with Collision Avoidance. When a node is ready to send a probe packet, it intercepts whether there is a co-channel carrier in the medium. If not, it means that the channel is idle and the node will go directly into the data-transmission state. If carrier exists, the node will detect the channel after a random back-off for a period of time. Here, we assume that self-learning starts at the beginning of a time slot. The duration of a self-learning cycle can be calculated as Eq. (10), where  $\varrho$  is a system parameter.

$$D_{LEA} = \sum_{i=1}^{\rho\vartheta} \sum_{j=1}^i 2Vd\varrho \left[ \frac{c^A}{c} \cdot \left(1 - \frac{c^A}{c}\right)^{\rho\vartheta-1} + \frac{c^A}{c} \cdot \sum_{k=1}^{\rho\vartheta-1} C_{\rho\vartheta-1}^k \cdot \left(1 - \frac{c^A}{c}\right)^k \right] \quad (10)$$

Here, the probability of any node waking up in each slot is  $c^A/c$  and the number of forwarders in forwarder set of node  $i$  is  $\rho\vartheta$ . In Eq. (10),  $\frac{c^A}{c} \cdot \left(1 - \frac{c^A}{c}\right)^{\rho\vartheta-1}$  represents the situation that the sender wakes up in a slot but all of its forwarders wake up in other slots. And  $\frac{c^A}{c} \cdot \sum_{k=1}^{\rho\vartheta-1} C_{\rho\vartheta-1}^k \cdot \left(1 - \frac{c^A}{c}\right)^k$  represents that the sender wakes up in a slot and most of its forwarders wake up in the same slot, except for  $k$  forwarders,  $k = 1, 2, \dots, \rho\vartheta - 1$ .

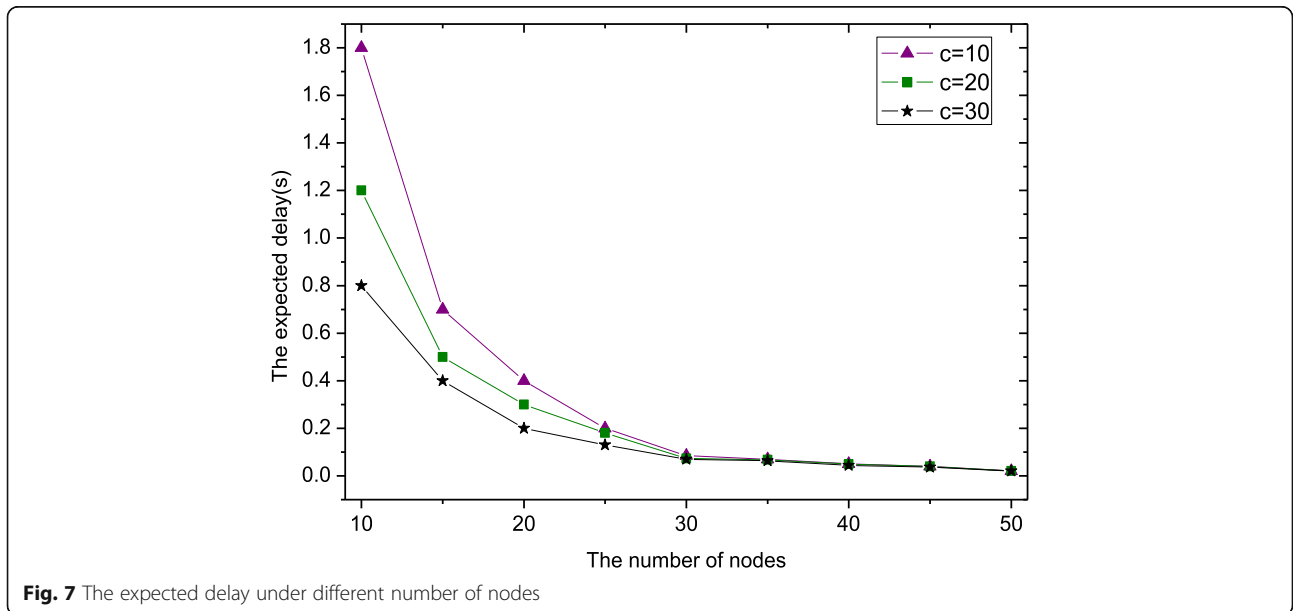


Because the whole self-learning process includes simultaneous learning among multiple regions, the time required to complete self-learning within a certain range (denoted as  $D_{LEA}^{TOT}$ ) is the maximum value of the self-learning time between the regions.

### 5 Performance analysis of LS approach

#### 5.1 Delay analysis

After the completion of self-learning, all nodes in any two regions separated by  $d$  maintain the synchronization of duty cycle. Compared with other methods, in which nodes need to wait for the



corresponding candidate forwarders to wake up before sending the packets because their duty cycle is not necessarily synchronous, this approach can optimize the performance of this respect, thus effectively reducing the sender wait time. Figure 6 shows one of the possible multi-hop path of the packet from the sender node A to the sink.

**Theorem 1** *The expected vertical distance from one of sender nodes to its forwarder is*

$$d_x = \int_0^x \frac{2x(d-x)}{\vartheta_x} \cdot \cos^{-1} \left( \frac{(d-x)^2 + d^2 - x^2}{2(d-x)d} \right) dx \quad (11)$$

*Proof* According to [4], the probability density function of the expected vertical distance from one of the sender nodes to its forwarder is  $\mathcal{F}_d(x) = \frac{2(d-x)}{\vartheta_x} \cdot \cos^{-1} \left( \frac{(d-x)^2 + d^2 - x^2}{2(d-x)d} \right)$ . The expected  $d_x$  can be calculated by the formula  $d_x = \int_0^x x \cdot \mathcal{F}_d(x) dx$ . ■

Although the duty cycle of most of the nodes in the forwarder set are synchronized with the sender's cycle, we prefer to select the relay node closer to the sink. Therefore, there is a need to reach a balance between the minimum delay and the maximum vertical distance.

**Theorem 2** *When packet forwarding is basing on LS approach, the duty cycle of nodes and their forwarders almost synchronous. When the distance from a node and the sink is L, the delay of end-to-end transmission can be calculated in Eq. (13).*

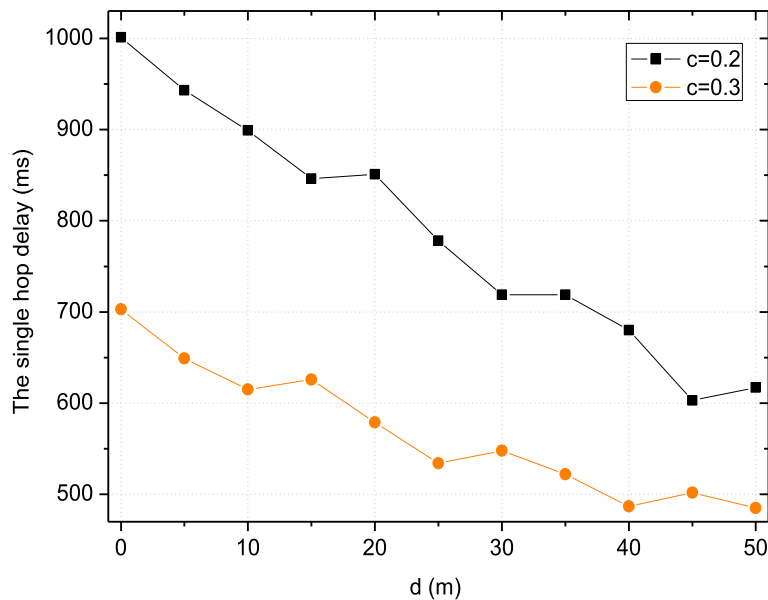
$$Y_{i-1,i} = \sum_{i=0}^{\frac{D_{COM}}{c}-2} i \left( 1 - \frac{c}{D_{COM}} \right)^{i\rho\vartheta} \left[ 1 - \left( 1 - \frac{c}{D_{COM}} \right)^{\rho\vartheta} \right] + \left( \frac{D_{COM}}{c} - 1 \right) \left( 1 - \frac{c}{D_{COM}} \right)^{\left( \frac{D_{COM}}{c} - 1 \right)\rho\vartheta} \quad (12)$$

$$Y_{ETE} = \left[ \frac{L}{d_x} \right] Y_{i-1,i} \quad (13)$$

*Proof*  $Y_{i-1,i}$  is the average delay of the transmission from  $(i-1)$ th to  $i$ th node. According to [4], the duty cycle of each node is  $c/D_{COM}$  in transmission and the probability density function of the delay in each slot is  $\left( 1 - \frac{c}{D_{COM}} \right)^{i\rho\vartheta} \left[ 1 - \left( 1 - \frac{c}{D_{COM}} \right)^{\rho\vartheta} \right]$ , where  $i = 0, 1, \dots, \frac{c}{D_{COM}} - 2$ . Here, the vertical distance is greater than or equal to  $x$ , so the expected hop count is  $\frac{L}{d_x}$ . Then, the total delay can be obtained by single hop delay and hop count. ■

Based on this formula, Fig. 7 shows the expected delay under different number of nodes. As we can see from the diagram, the overall trend is that the more the number of forwarders is, the shorter the delay is.

And as shown in Fig. 8, with the increase of single hop distance, the single hop delay is reduced. This indicates that the longer distance can lead to the shorter delay. At the same time, as duty cycle increases, the delay will decrease.



**Fig. 8** The single hop delay under different single hop distance

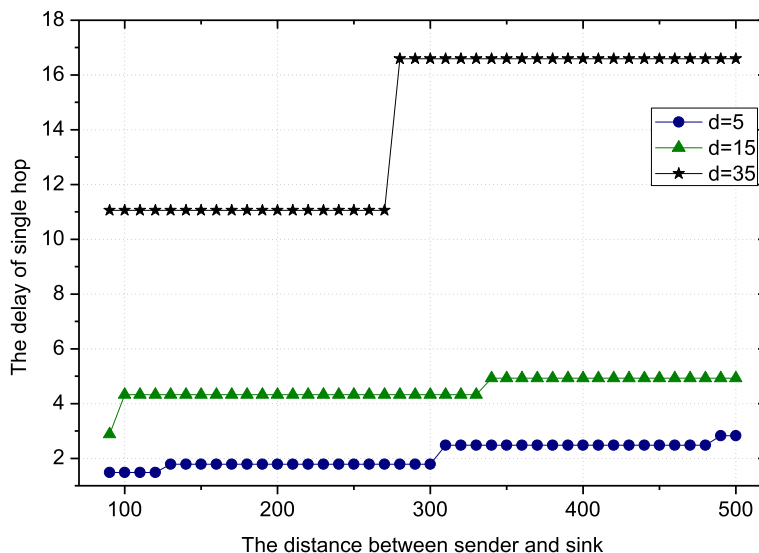


Fig. 9 The single hop delay under different distance

As Fig. 9 shows, the one-hop delay increases with the increase of the distance  $L$ . Meanwhile, the distance between a sender and its forwarder also reflects one-hop delay: the longer the one-hop distance, the longer the one-hop delay.

5.2 Energy consumption analysis

The network lifetime is determined by energy consumption, and the selection of forwarders is also related to residual energy. In LS approach, nodes have four types of states: transmission of packets, reception of packets, self-learning, and low power

listening. In a general way, while the number of packets in network is large, the energy they consume is serious because the packet transmission and reception are the main energy-consuming operation. Beyond that, in this approach, the self-learning process is also an operation that consumes a lot of energy. On the other hand, the overhead of time slot can be divided into three parts: when there have data packet transmission, listening, and sleeping time. This is included in the calculation of the above four parts of the energy consumption. The total energy consumption can be calculated in the following equation

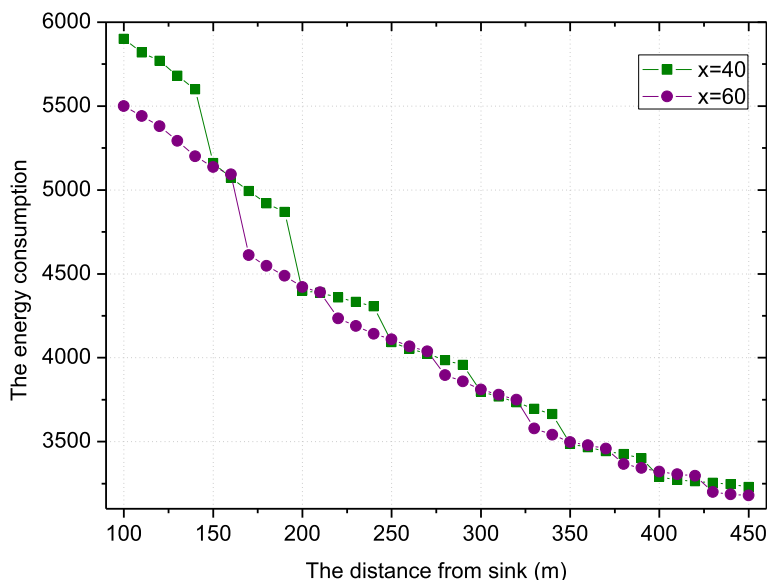


Fig. 10 The energy consumption under different x values

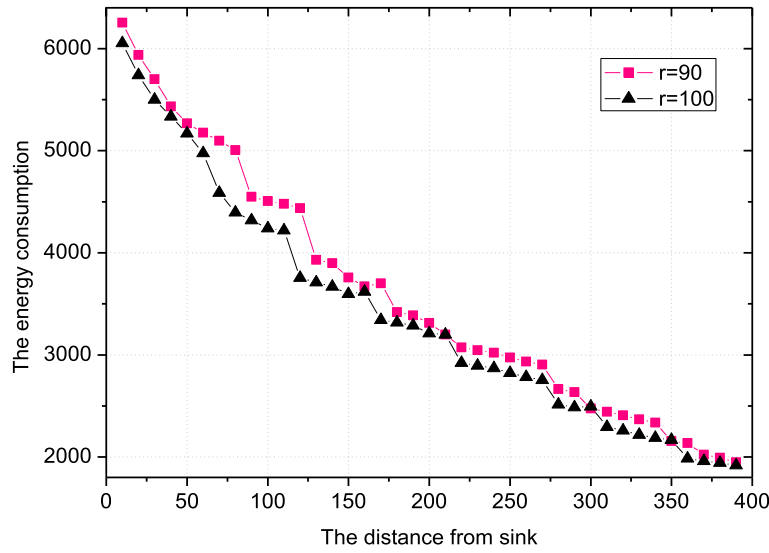


Fig. 11 The energy consumption under different duty cycle

$$E_U = E_T N_T + E_R N_R + E_{LPL} D_{COM} + E_{LEA} D_{LEA} \quad (14)$$

where  $E_U$  is the total energy consumption,  $E_T$  is the energy consumption of a packet transmission,  $E_{LPL}$  represents the energy consumption in low power listening, and  $E_{LEA}$  is the energy consumption in self-learning.  $N_T$  and  $N_R$  are the number of packets in transmission and the packet amount in reception respectively.  $D_{COM}$  and  $D_{LEA}$  are the duration of communication and self-learning.

According to [4], assuming that the radius of network is  $D$ , the transmission range of each node is  $r$  and the distance between the sender and sink is  $l$ ;  $N_T$  and  $N_R$  can be calculated by following equations in which  $i$  is an integer here.

$$N_R = \frac{r(i+1)i}{2l} + (i+1) \mid ir + l < D \quad (15)$$

$$N_T = N_R + 1 \quad (16)$$

The energy consumption in a packet sending includes consumption of the real packet transmission and the related preamble consumption.  $\epsilon_T D_P$  represents the energy consumption of packet transmission. The preamble transmission is a periodic operation in the whole transmission, which can inform the relay node of the packet's arrival. The average energy consumption in a packet transmission can be expressed as

$$E_T = \epsilon_T D_P + (\epsilon_T D_B + \epsilon_R D_{ACK}) \frac{(1 - \frac{c^A}{c}) D_{COM}}{2(D_B + D_{ACK})} \quad (17)$$

The second part of Eq. (17) represents the power consumption of periodic preamble transmission.

The energy consumption in packet reception can be calculated as

$$E_R = \epsilon_R D_P + \epsilon_T D_{ACK} + \epsilon_R D_B \quad (18)$$

where  $\epsilon_R D_P$  represents the power consumption of the packet reception,  $\epsilon_T D_{ACK}$  is the power consumption in the transmission of acknowledgment message, and  $\epsilon_R D_B$  represents the power consumption of the preamble reception.

The energy consumption in self-learning can be expressed by the following equation:

$$E_{LEA} = \epsilon_{LEA} D_{LEA} = D_{LEA} (\epsilon_R + \epsilon_T + \epsilon_R) \quad (19)$$

The energy consumption used for low power listening of the node from sink  $L$  meters (denoted as  $E_{LPL}^L$ ) can be calculated as

Table 2 Simulation parameters

Parameter	Value	Description
$E_O$	0.5	Initial energy (J)
$D_{COM}$	100	Communication duration (ms)
$\epsilon_T$	0.0511	Power consumption of transmission (W)
$\epsilon_R$	0.0588	Power consumption of reception (W)
$\epsilon_S$	$2.4 \times 10^{-7}$	Power consumption of sleeping (W)
$r$	80	The transmission range of a node (m)
$D_B$	0.26	Preamble duration (ms)
$D_{ACK}$	0.26	Acknowledge window duration (ms)
$D_P$	0.93	Packet duration (ms)



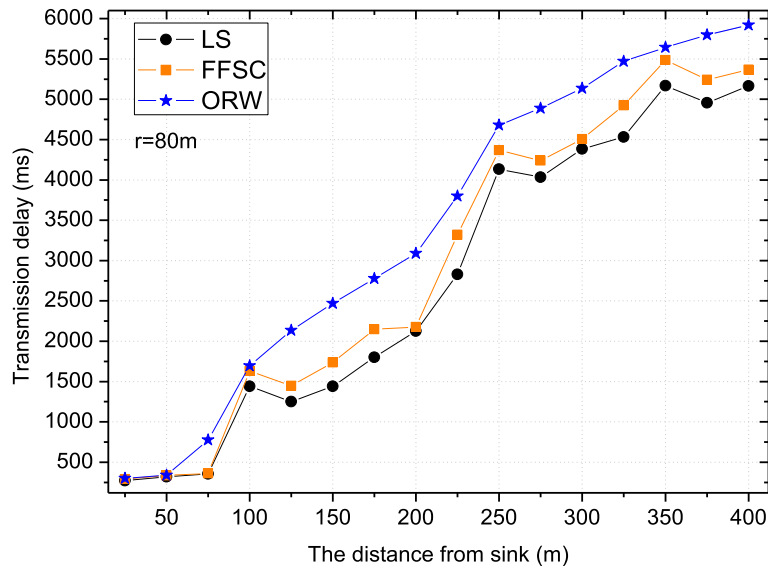


Fig. 12 The total delay under different distance from sink

$$E_{LPL}^L = \epsilon_S \left(1 - \frac{c^A}{c}\right) + \epsilon_R \frac{c^A}{c} - \psi_T^L - \psi_R^L \quad (20)$$

$$\psi_T^L = \left\{ \left[ \frac{T_{COM} \left(1 - \frac{c^A}{c}\right)}{2} + D_B + D_{ACK} \right] \epsilon_S + \epsilon_R D_B \right\} \frac{N_T^L}{T_{COM}} \quad (21)$$

The process of low power listening consists of two parts: listening and sleeping states, which correspond to the first and second part in Eq. (20). Because the energy consumption of packet transmission and reception is calculated in the equation before,  $\psi_T^L$  and  $\psi_R^L$  should be subtracted from the calculation for  $E_{LPL}^L$ .

$$\psi_R^L = [\epsilon_R D_B + (D_P + D_{ACK}) \epsilon_S] \frac{N_R^L}{T_{COM}} \quad (22)$$

According to [5],  $\psi_T^L$  and  $\psi_R^L$  can be expressed by Eqs. (21) and (22) respectively:

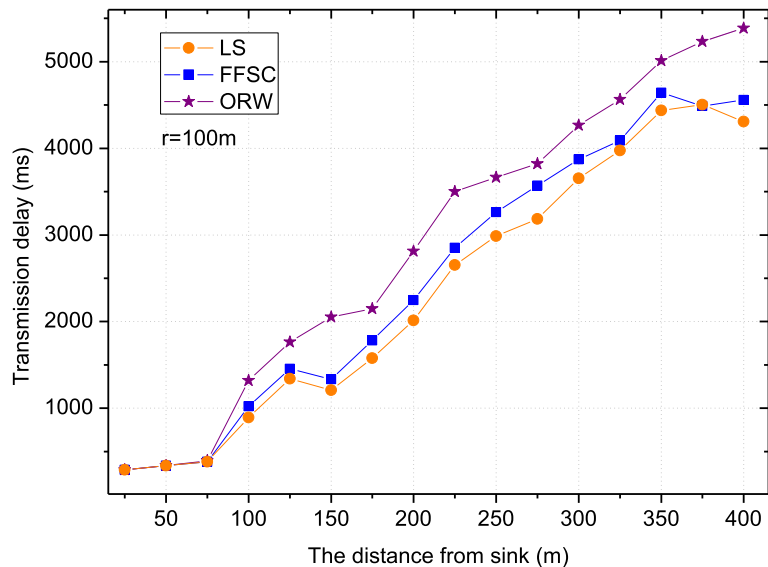


Fig. 13 The total delay under different distance from sink

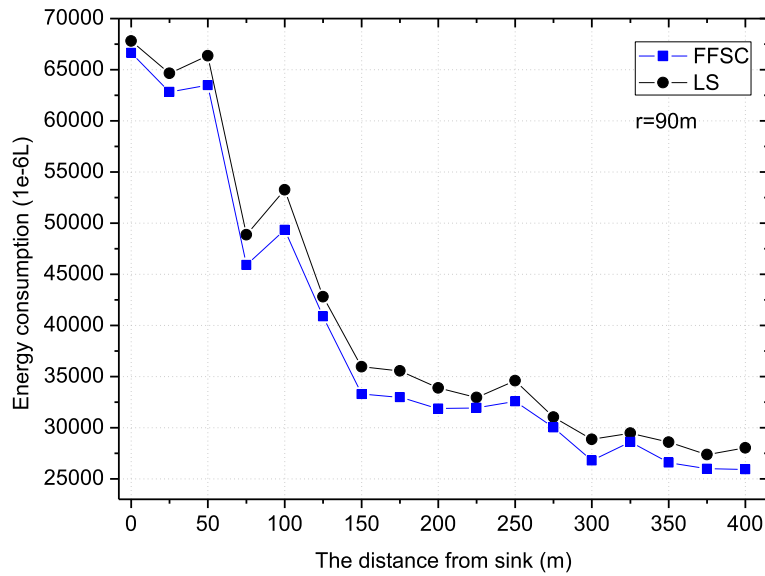


Fig. 14 The energy consumption under different distance from sink

Based on the analysis above, the energy consumption of nodes which are  $x$  meters away from the corresponding relay nodes can be expressed as

$$E_U^x = E_T^x N_T + E_R^x N_R + E_{LPL}^x D_{COM}^x + E_{LEA}^x D_{LEA}^x \quad (23)$$

Because the range of the forwarder set is  $x$ , and the average distance from the sender and the region of relay nodes is  $d$ , the total energy consumption can be calculated as follows:

$$E_U = \left[ \frac{L}{d} \right] \left( E_T^x N_T + E_R^x N_R + E_{LPL} D_{COM} + E_{LEA}^x D_{LEA} \right) \quad (24)$$

**Theorem 3** In the approach presented in this paper, suppose that the initial energy of node  $i$  is  $E_O^i$  and there are  $n$  nodes in the network, the lifetime of network can be expressed as

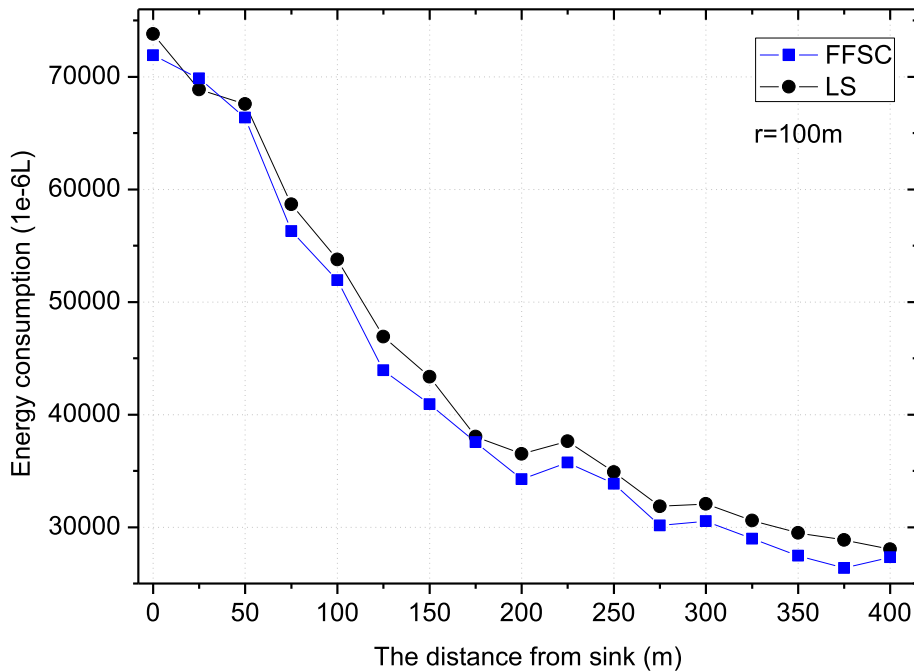


Fig. 15 The energy consumption under different distance from sink

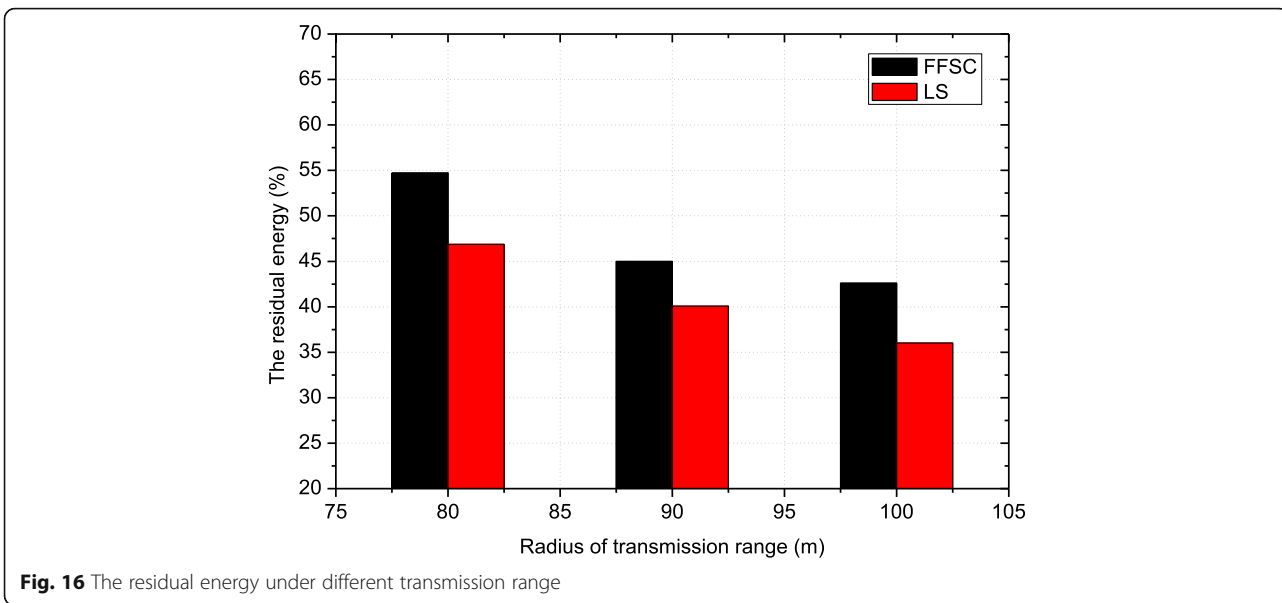


Fig. 16 The residual energy under different transmission range

$$T_N = \frac{E_O^i}{\max_{1 \leq i \leq n} (E_T N_T^i + E_R N_R^i + E_{LPL} D_{COM} + E_{LEA} D_{LEA})} \tag{25}$$

*Proof* Network lifetime is defined as the period before the node which has the largest energy dies out due to its energy drain. In other words, the lifetime of the node with the maximum energy and the initial energy refer to the network lifetime.  $E_O^i$  is the initial energy of the  $i$ th node, and the largest energy consumption of the node is  $\max_{1 \leq i \leq n} (E_T N_T^i + E_R N_R^i + E_{LPL} D_{COM} + E_{LEA} D_{LEA})$ . ■

Based on Eq. (24), Fig. 10 shows the energy consumption under the different  $x$  values. It can be observed that

the larger the  $x$  is, the smaller the energy consumption is, which is because the hop count decreases with the increase of  $x$ . Meanwhile, as the distance between the sender and sink increases, the energy consumption of nodes are also reduced.

As shown in Fig. 11, the trend of energy consumption changes with the total distance from sink is the same as in Fig. 10. In the approach proposed in this paper, when each node becomes the sender, it is taken as the center in the network. In the transmission of packets, the selection of relays in each packet forwarding always follow the principle of selecting nodes that are closer to the sink. Thus, the closer the distance between a node and sink, the greater the energy consumed, which can be observed from Fig. 11.

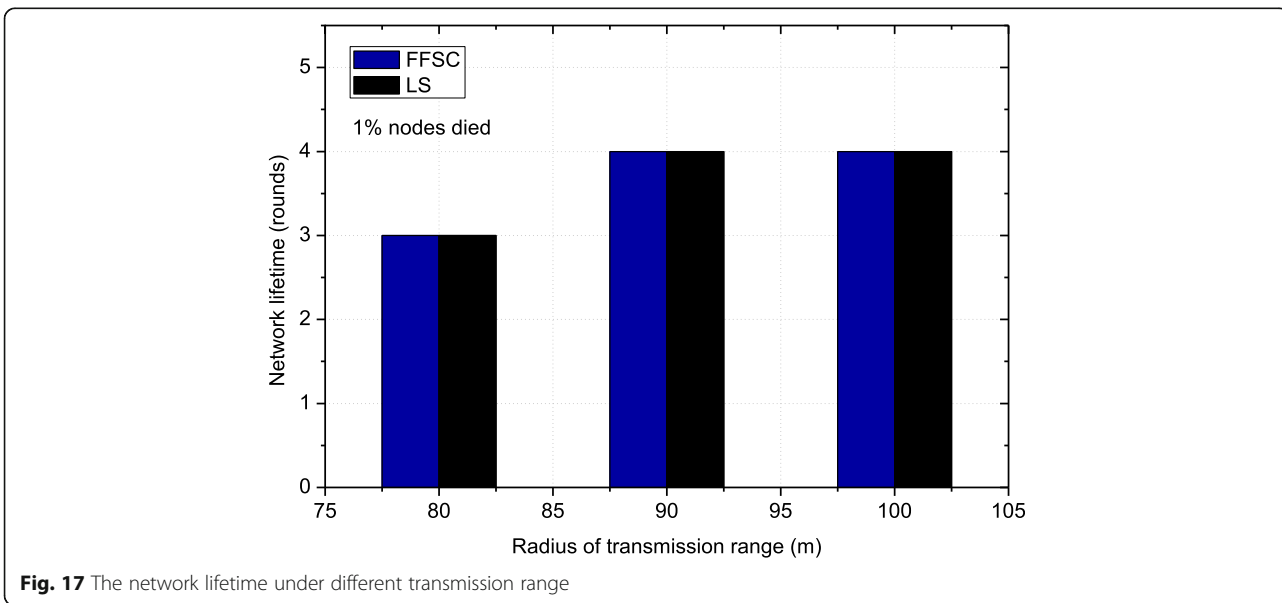


Fig. 17 The network lifetime under different transmission range

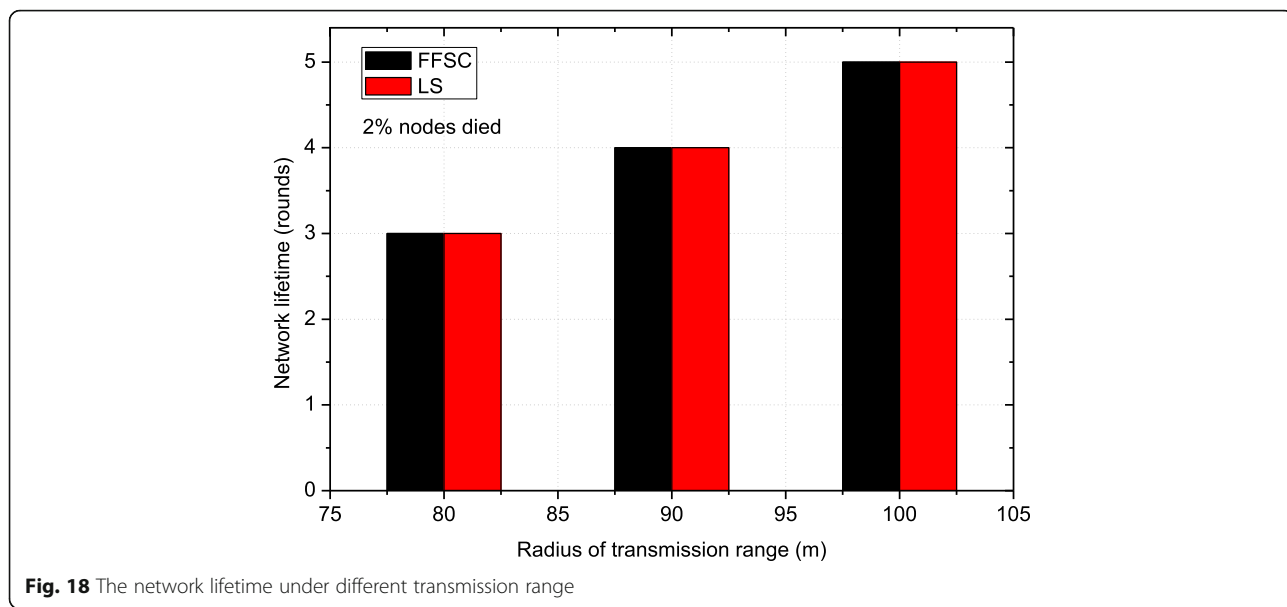


Fig. 18 The network lifetime under different transmission range

### 6 Simulation result

The performance of LS approach by simulation is evaluated in this chapter. The reason for choosing FFSC approach to compare is that the main difference between it and the protocol we proposed is that FFSC approach does not have self-learning stage and self-learning is the innovation of our protocol. The whole evaluation system includes the following main aspects: transmission delay, energy consumption, and network lifetime. Except that we analyze the effect of some parameters, such as the node density, the radius of the communication range, and the duty cycle. When there is no special explanation, the transmission range of a node (denoted as  $r$ ) is set to 80 m. In this simulation, when the location of each node

is randomly generated, the forwarding nodes set of the sender is selected by the method in 4.2. And  $n$  sensors are deployed here,  $n = 700$  without special explanation. The analysis results show that compared with FFSC approach, the LS approach can reduce the total transmission delay by 5.13–11.64% and increase the energy efficiency by 14.29–17.53%, while guaranteeing the network lifetime is not less than other approaches. The values of the main simulation parameters are listed in Table 2.

#### 6.1 Transmission delay

Figures 12 and 13 show the total transmission delay of the LS, FFSC, and ORW under different distance from sink. All nodes have the same duty cycle in ORW

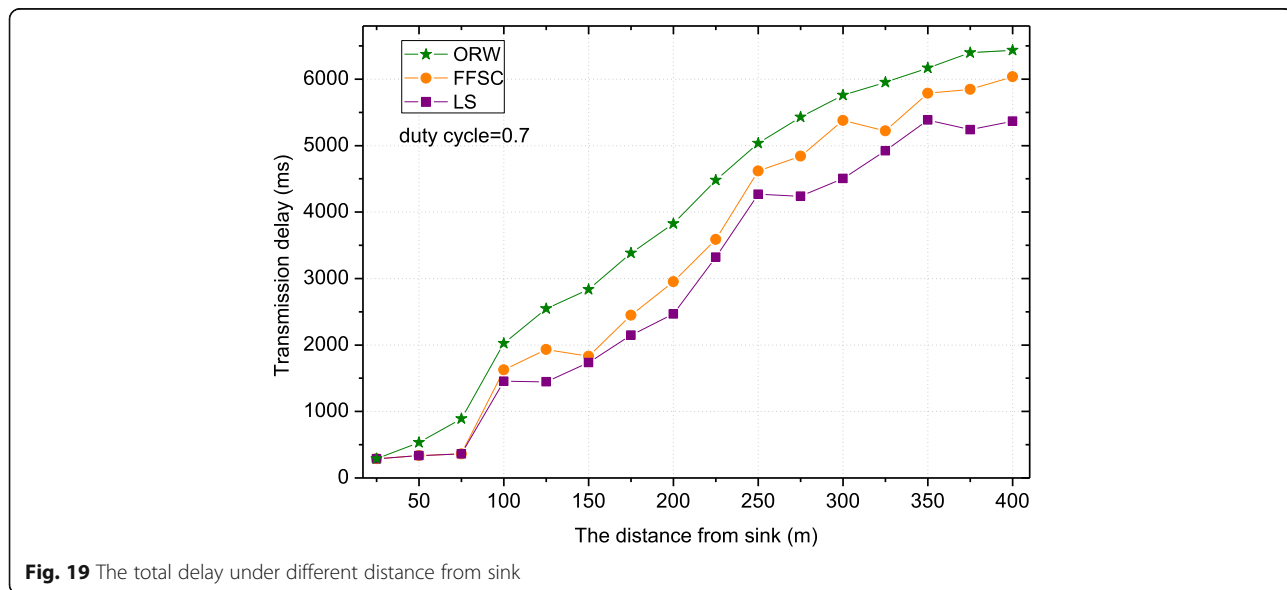


Fig. 19 The total delay under different distance from sink

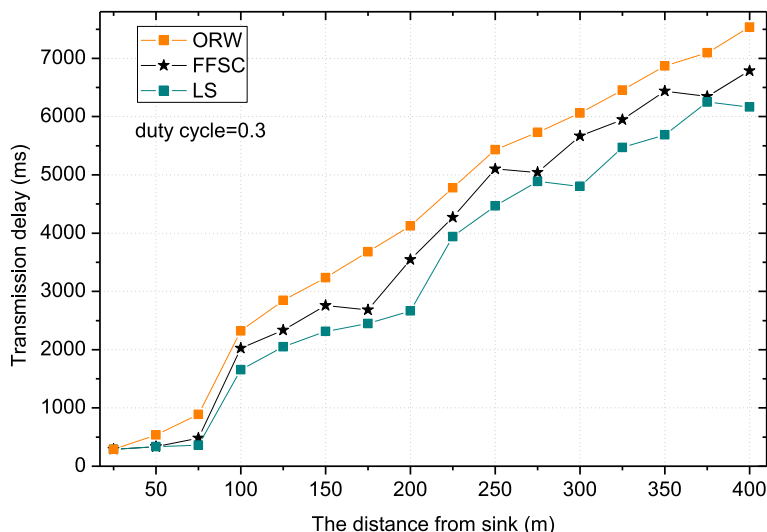


Fig. 20 The total delay under different distance from sink

approach. Therefore, the distance between sender node and sink is mainly related to the total delay. As shown in the first curve in Fig. 12, the greater the distance, the greater the total delay. Meanwhile, because ORW does not consider the location of nodes, there exists a problem that the forwarder far away from the sink may be selected. So the total delay of ORW approach is obviously longer than the other two approaches. Similarly, in FFSC approach, the communication duty cycle of one node is as same as others. The difference between ORW and FFSC lies in the forwarder selection. With the consideration of the location of nodes, candidate forwarders are delineated within a range closer to the sink. Thus, compared to ORW, the total delay of FFSC approach is shorter. Unlike

the above two approach, LS has been improved on the basis of FFSC, which is mainly embodied in self-learning. In the self-learning cycle, the duty cycle of the sender and its corresponding forwarder set are synchronized. It successfully reduces the total delay of transmission. Compared with FFSC, the LS approach can reduce the transmission delay by 5.13–11.64%, while the network lifetime of these two protocols are the same. Thus, it can be seen that the approach proposed in this paper is better than other approaches in terms of transmission delay.

6.2 Energy efficiency and network lifetime

The energy utilization can reflect the performance of the protocol and also affect the selection of forwarding

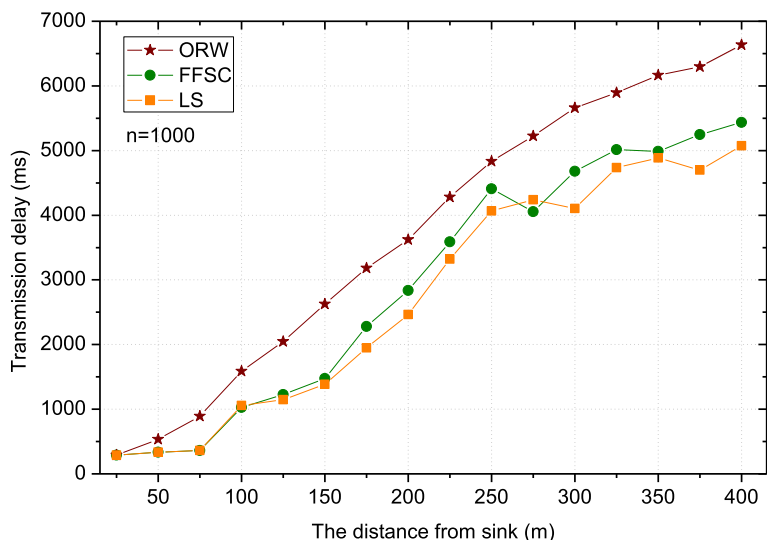


Fig. 21 The total delay under different distance from sink



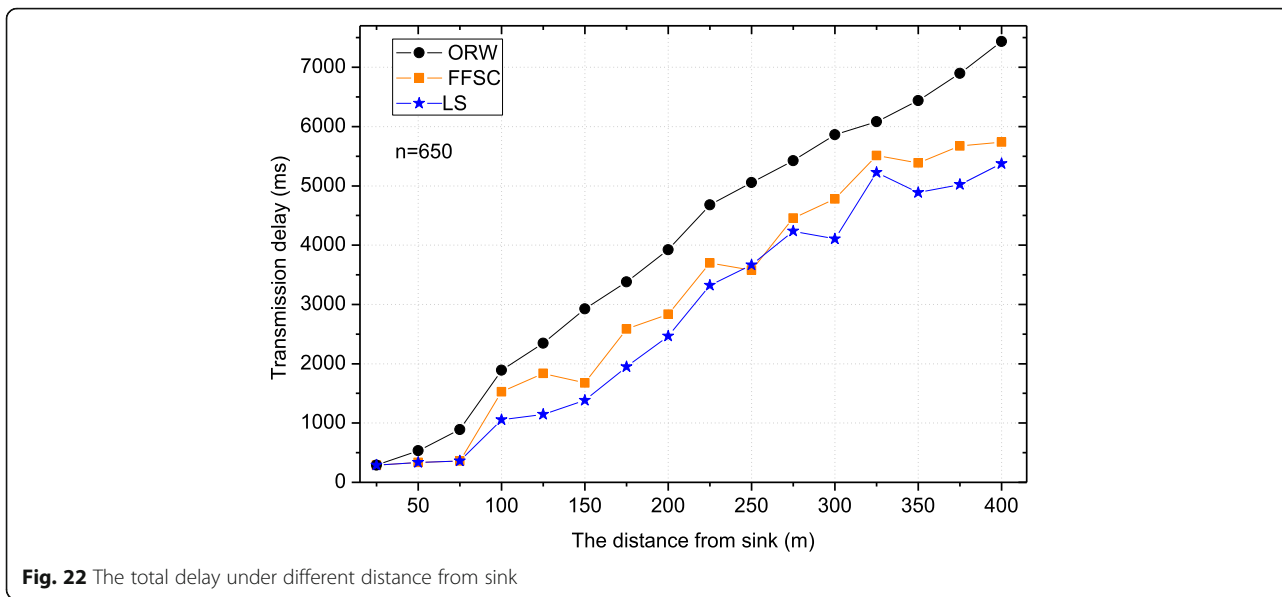


Fig. 22 The total delay under different distance from sink

nodes. Meanwhile, the network lifetime is related to energy consumption of nodes. In this section, the performance of LS approach in energy consumption and network lifetime is analyzed.

Figures 14 and 15 show the energy consumption of the LS and FFSC approach under different distance from sink. Here, the vertical distance between a sender and its forwarder has to be greater than or equal to  $x$ , while  $x$  is related to other parameters in the network. As can be seen in the following figures, the longer the distance from the sink, the less the energy consumed. Compared with the FFSC approach, LS consumes energy efficiently in non-hotspot areas. By comparing the two pairs of curves in Figs. 14 and 15, it can be observed that the

maximum energy consumption of LS and FFSC is relatively close, which is partly due to the large energy consumption during self-learning period. The bell-shaped is presented in Fig. 14 because there is no need to do self-learning before transmission when the sender is close to the sink, so at the beginning of the curve, the energy consumption of the two approaches is similar. And with the distance from the sink increases, because the partial synchronization is achieved in the self-learning period, the energy consumption during the transmission is effectively reduced. Thus, the bell-shaped is formed in Fig. 14.

And the trends of curves in Figs. 14 and 15 prove that the closer the distance between a node and sink, the

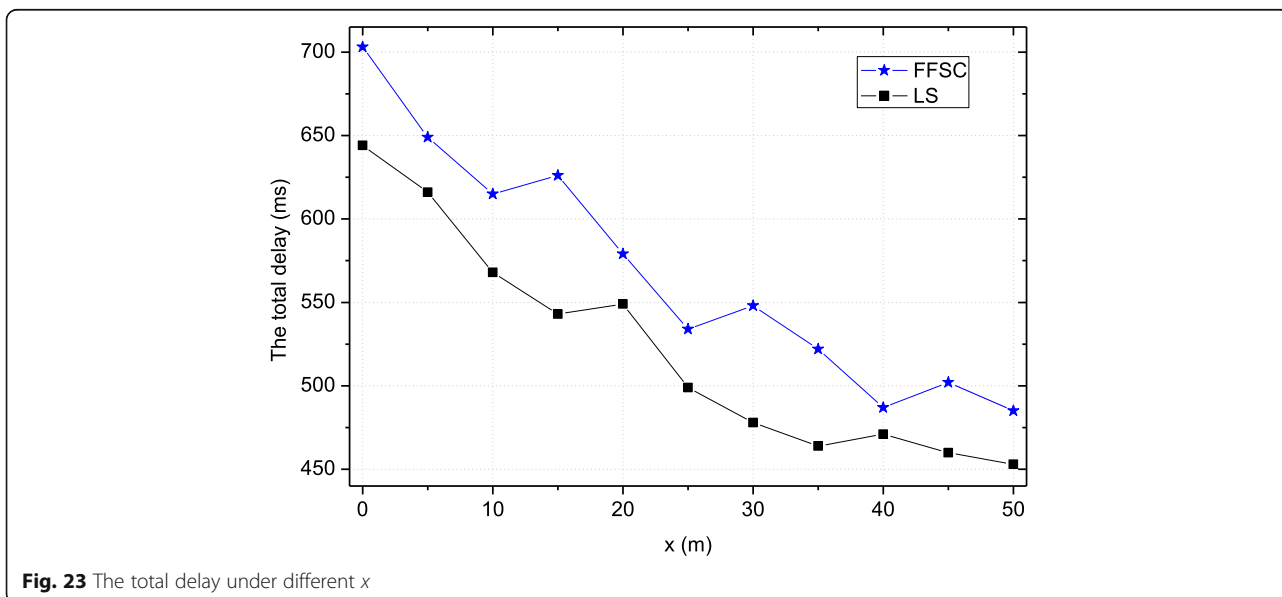


Fig. 23 The total delay under different x

greater the energy consumed. This trend is the same as in Fig. 11.

Figure 16 shows the column diagram of residual energy under different radius of transmission range. Compared with FFSC approach, the LS can increase the efficiency of energy utilization by 14.29–17.53%, which can be observed in the figure. In a word, while adopting the LS approach, the energy utilization reaches a better balance. Especially for the areas farther away from the sink, the energy efficiency of nodes has improved.

Figures 17 and 18 show the network lifetime of FFSC and LS approach under different number of the first nodes that use up their energy. Compared to FFSC approach, although the efficiency of energy utilization during the transmission is improved, the network lifetime of the approach proposed in this paper is unchanged. This is because that the energy saved during the transmission compared to FFSC approach is almost offset by the energy consumption during the self-learning period.

### 6.3 Effects of other parameters on the performance

In this section, we analyze the performance of LS approach which is affected by other factors in the network. The length of a duty cycle is an important factor influencing the end-to-end delay. We usually expect a near perfect duty cycle to promote the probability that the sender and its candidate relays will be active at the same time. Figures 19 and 20 show the total transmission delay under the different duty cycle. As they shows, compared with other approach, the total delay of the approach proposed in this paper has been reduced, which also proves the correctness of our ideas. At the same time, it can be observed that the larger the duty cycle is,

the shorter the total delay is. This is the same as the conclusion obtained in Fig. 8.

In addition to duty cycle, the number of nodes is also one of the factors which we are concerned with. Figures 21 and 22 show the total delay of the different approaches under different number of nodes. Compared with ORW and FFSC, LS approach can achieve smaller total delay than others. This is because that in ORW and FFSC approach, the nodes in the forwarder set are not synchronized with the corresponding sender. So if the number of nodes increases, the sender wait time before its forwarding nodes wake up will decrease and the delay of each hop in transmission will also decrease. This trend is the same as in Fig. 7. And in LS approach, due to the synchronization between the sender and its candidate forwarding nodes, the sender does not require to wait for the receiver to wake up or only need to wait for a shorter time than without synchronization.

The parameter  $d$  mentioned in the previous section also has an effect on transmission delay. In general, the longer the one-hop distance, the less the number of forwarders that a packet will pass. As can be seen in Fig. 23, LS approach is less than the FFSC in the total transmission delay, and as  $d$  becomes longer, the total delay decreases. On the contrary, when using LS approach, the one-hop delay under different  $x$  grows with the increase of  $x$ , as can be seen in Fig. 24.

### 7 Conclusions

To reduce delay is important for the Industrial Internet of Things (IIoTs). However, previous strategies often require sacrificing some other network performance to reduce delay. In general, the reduction of delay is

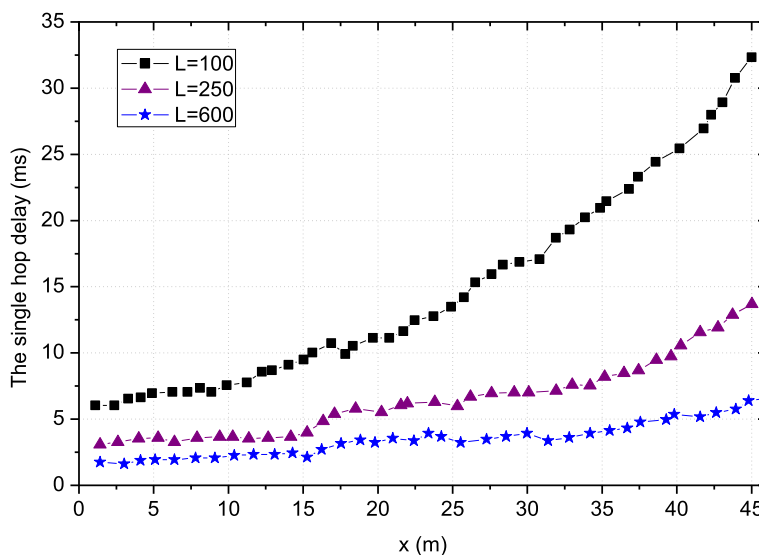


Fig. 24 The total delay under different x

achieved by consuming a certain amount of energy and reducing the network lifetime. In our LS scheme, a novel approach is proposed, which can effectively reduce delay without reducing network lifetime. The main innovation of this approach is that partial synchronization strategy is used to keep the sender synchronized with its forwarding nodes, thus greatly reducing the difficulty of maintaining synchronization and effectively reducing the delay. Moreover, this synchronization is implemented by self-learning method and has strong robustness. In addition, the synchronous energy consumption takes full advantage of the remaining energy in the network, so as to improve the energy efficient utilization without reducing the network lifetime. The LS approach proposed in this paper has a wide range of applicability and not only can be applied to IIoTs but also can be widely applied to IoT-based edge network. That is what we will research in the future.

#### Abbreviation

ACK: Acknowledgement; AWP: Adaptable Wakeup Period; DC: Data center; EDC: Expected Duty Cycled Wakeups; FFSC: A Fast data collection for the node Faraway sink and Slow data collection for the node Close to sink; FNs: The sets of the candidate forwarding nodes of the senders; IIoTs: The Industrial Internet of Things; IoT: The Internet of Things; LS: Learning-based synchronous; M2M: Machine-to-machine; MAC: Media Access Control; ORW: Opportunistic Routing in Wireless sensor networks; SDC: Same duty cycle; TDMA: Time division multiple access; WSNs: Wireless Sensor Networks

#### Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (61772554, 61379110, 61572526, 61572528) and the National Basic Research Program of China (973 Program) (2014CB046305).

#### Funding

Not applicable.

#### Availability of data and materials

Not applicable.

#### Authors' contributions

MW is the main author of the current paper. AL contributed to the conception and design of the study. YW, XL, MM, and MZ commented the work. All authors read and approved the final manuscript.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

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

#### Author details

<sup>1</sup>School of Information Science and Engineering, Central South University, 401 room, Computer building, Changsha 410083, China. <sup>2</sup>Hunan University of Commerce, Changsha 410205, China. <sup>3</sup>Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA. <sup>4</sup>School of Software, Central South University, Changsha 410075, China.

Received: 27 September 2017 Accepted: 19 December 2017

Published online: 09 January 2018

#### References

- Xiao F, Sha L T, Yuan Z P, Wang R C. VulHunter: a discovery for unknown bugs based on analysis for known patches in Industry Internet of Things, *IEEE Transactions on Emerging Topics in Computing*, 2017, 1-13.
- H Zhu, F Xiao, L Sun, R Wang, P Yang, R-TTWD: robust device-free through-the-wall detection of moving human with WiFi. *IEEE Journal on Selected Areas in Communications* **35**(5), 1090–1103 (2017)
- A Liu, Z Chen, N Xiong, An adaptive virtual relaying set scheme for loss-and-delay sensitive WSNs. *Inf. Sci.* **424**, 118–136 (2018)
- Y Liu, A Liu, Y Hu, Z Li, YJ Choi, H Sekiya, J Li, FFSC: an energy efficiency communications approach for delay minimizing in Internet of things. *IEEE Access* **4**, 3775–3793 (2016)
- Liu, X., Zhao, S., Liu, A., Xiong, N., & Vasilakos, A. V. Knowledge-aware proactive nodes selection approach for energy management in Internet of things. *Future generation computer systems*, 2017, <https://doi.org/10.1016/j.future.2017.07.022>.
- Xu, Y., Liu, A., Huang, C. Delay-aware program codes dissemination scheme in Internet of everything. *Mobile Information Systems*, 2016. <https://doi.org/10.1155/2016/2436074>.
- Chen, X., Ma, M., Liu, A. Dynamic power management and adaptive packet size selection for IoT in e-Healthcare. *Computers & Electrical Engineering*, 2017, <https://doi.org/10.1016/j.compeleceng.2017.06.010>.
- S Andreev, O Galinina, A Pyattaev, M Gerasimenko, T Tirronen, J Torsner, Y Koucheryavy, Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap. *IEEE Commun. Mag.* **53**(9), 32–40 (2015)
- Kawabata, H., Ishibashi, K., Vuppala, S., de Abreu, G. T. Robust relay selection for large-scale energy-harvesting IoT networks. *IEEE Internet of Things Journal*, 2017, 4(2), 384–392.
- H Zhang, P Cheng, L Shi, J Chen, Optimal DoS attack scheduling in wireless networked control system. *IEEE Trans. Control Syst. Technol.* **24**(3), 843–852 (2016)
- Duan, J., Gao, D., Yang, D., Foh, C. H., Chen, H. H. An energy-aware trust derivation scheme with game theoretic approach in wireless sensor networks for IoT applications. *IEEE Internet of Things Journal*, 2014, 1(1), 58–69.
- PV Mekikis, A Antonopoulos, E Kartsakli, L Alonso, C Verikoukis, in *Global Communications Conference (GLOBECOM)*. Connectivity analysis in wireless-powered sensor networks with battery-less devices (2016), pp. 1–6
- A Liu, Q Zhang, Z Li, YJ Choi, J Li, N Komuro, A green and reliable communication modeling for industrial internet of things. *Computers & Electrical Engineering* **58**, 364–381 (2017)
- HW Kim, JH Park, YS Jeong, Sustainable load-balancing scheme for inter-sensor convergence processing of routing cooperation topology. *Sustainability* **8**(5), 436 (2016)
- Y Sung, YS Jeong, JH Park, Beacon-based active media control interface in indoor ubiquitous computing environment. *Clust. Comput.* **19**(1), 547–556 (2016)
- Z Wang, F Xiao, N Ye, R Wang, P Yang, A see-through-wall system for device-free human motion sensing based on battery-free RFID. *ACM Transactions on Embedded Computing Systems (TECS)* **17**(1), 6 (2017)
- Chen, Z., Liu, A., Li, Z., Choi, Y. J., Sekiya, H., & Li, J. Energy-efficient broadcasting scheme for smart industrial wireless sensor networks. *Mobile Information Systems*, 2017, <https://doi.org/10.1155/2017/7538190>
- Liu, X., Dong, M., Ota, K., Yang, L. T., & Liu, A. Trace malicious source to guarantee cyber security for mass monitor critical infrastructure. *Journal of Computer and System Sciences*, 2016. <https://doi.org/10.1016/j.jcss.2016.09.008>
- A Liu, X Liu, T Wei, LT Yang, SC Rho, A Paul, Distributed multi-representative re-fusion approach for heterogeneous sensing data collection. *ACM Trans Embed Comput Syst.* **16**, 73 (2017). <https://doi.org/10.1145/2974021>
- Y Liu, A Liu, S Guo, Z Li, YJ Choi, H Sekiya, Context-aware collect data with energy efficient in cyber-physical cloud systems. *Futur. Gener. Comput. Syst.* (2017). <https://doi.org/10.1016/j.future.2017.05.029>
- Liu, Q., Liu, A. On the hybrid using of unicast-broadcast in wireless sensor networks. *Computers & Electrical Engineering*, 2017, <https://doi.org/10.1016/j.compeleceng.2017.03.004>
- Liu X., Liu, A., Deng Q., Liu, H. Large-scale programing code dissemination for software defined wireless networks. *Computer Journal*. **60**(10), 1417–1442 (2017).

23. X Liu, Y Liu, H Song, A Liu, Big data orchestration as a service networking. *IEEE Commun. Mag.* **55**(9), 94–101 (2017)
24. X Chen, Y Xu, A Liu, Cross layer design for optimal delay, energy efficiency and lifetime in body sensor networks. *Sensors* **17**(4), 900 (2017). <https://doi.org/10.3390/s17040900>.
25. T Li, A Liu, C Huang, A similarity scenario-based recommendation model with small disturbances for unknown items in social networks. *IEEE Access* **4**, 9251–9272 (2016)
26. X Liu, G Li, S Zhang, A Liu, Big program code dissemination scheme for emergency software-define wireless sensor networks. *Peer-to-Peer Networking and Applications* (2017). <https://doi.org/10.1007/s12083-017-0565-6>
27. A Antonopoulos, C Verikoukis, Network-coding-based cooperative ARQ medium access control protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks* **8**(1), 601321 (2011)
28. PV Mekikis, A Antonopoulos, E Kartsaki, AS Lalos, L Alonso, C Verikoukis, Information exchange in randomly deployed dense WSNs with wireless energy harvesting capabilities. *IEEE Trans. Wirel. Commun.* **15**(4), 3008–3018 (2016)
29. A Liu, X Liu, Z Tang, LT Yang, Z Shao, Preserving smart sink location privacy with delay guaranteed routing scheme for WSNs. *ACM Trans. Embed. Comput. Syst.* **16**, 68 (2017). <https://doi.org/10.1145/2990500>
30. C Huang, M Ma, Y Liu, A Liu, Preserving source location privacy for energy harvesting WSNs. *Sensors* **17**, 724 (2017). <https://doi.org/10.3390/s17040724>
31. X Liu, A Liu, Z Li, S Tian, YJ Choi, H Sekiya, J Li, Distributed cooperative communication nodes control and optimization reliability for resource-constrained WSNs. *Neurocomputing* **270**, 122–136 (2017)
32. Z Chen, A Liu, Z Li, YJ Choi, J Li, Distributed duty cycle control for delay improvement in wireless sensor networks. *Peer-to-Peer Networking and Applications*. **10**, 559–578 (2017)
33. J Wang, A Liu, S Zhang, *Key parameters decision for cloud computing: insights from a multiple game model, concurrency and computation: practice and experience* (2017). <https://doi.org/10.1002/cpe.4200>
34. J Wang, A Liu, T Yan, Z Zeng, *A resource allocation model based on double-sided combinational auctions for transparent computing, peer-to-peer networking and applications* (2017). <https://doi.org/10.1007/s12083-017-0556-6>
35. Li, T., Liu, Y., Gao, L., Liu, A. A cooperative-based model for smart-sensing tasks in fog computing. *IEEE Access*, 2017, <https://doi.org/10.1109/ACCESS.2017.2756826>
36. Liu, Y., Liu, A., Li, Y., Li, Z., Choi, Y. J., Sekiya, H., & Li, J. APMD: a fast data transmission protocol with reliability guarantee for pervasive sensing data communication. *Pervasive and Mobile Computing*. <https://doi.org/10.1016/j.pmcj.2017.03.012>, 2017.
37. Zhang, Q., Liu, A. An unequal redundancy level based mechanism for reliable data collection in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*. **2016**(1), 258 (2016). <https://doi.org/10.1186/s13638-016-0754-6>.
38. Y Xu, X Chen, CH A Liu, A latency and coverage optimized data collection scheme for smart cities based on vehicular ad-hoc networks. *Sensors* **17**(4), 888 (2017). <https://doi.org/10.3390/s17040888>
39. M Dong, K Ota, A Liu, RMER: reliable and energy-efficient data collection for large-scale wireless sensor networks. *IEEE Internet of Things Journal* **3**(4), 511–519 (2016)
40. M Dong, K Ota, LT Yang, A Liu, M Guo, LSCD: a low-storage clone detection protocol for cyber-physical systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **35**(5), 712–723 (2016)
41. M Dong, K Ota, A Liu, M Guo, Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* **27**(1), 225–236 (2016)
42. KP Naveen, A Kumar, Relay selection for geographical forwarding in sleep-wake cycling wireless sensor networks. *IEEE Trans. Mob. Comput.* **12**(3), 475–488 (2013)
43. JH Lee, A traffic-aware energy efficient scheme for WSN employing an adaptable wakeup period. *Wirel. Pers. Commun.* **71**(3), 1879–1914 (2013)
44. CJ Liu, P Huang, L Xiao, TAS-MAC: a traffic-adaptive synchronous MAC protocol for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* **12**(1), 1 (2016)
45. SC Huang, PJ Wan, CT Vu, et al., Nearly constant approximation for data aggregation scheduling in wireless sensor networks. *IEEE INFOCOM*, 366–372 (2007). <https://doi.org/10.1109/INFCOM.2007.50>
46. Yu, B., Li, J., Li, Y. Distributed data aggregation scheduling in wireless sensor networks, *Proceedings of the IEEE INFOCOM*, 2009, 2159–2167. <https://doi.org/10.1109/INFCOM.2009.5062140>.
47. XH Xu, XY Li, X Mao, et al., A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* **22**(1), 163–175 (2011)
48. H Zhang, Y Shu, P Cheng, J Chen, Privacy and performance trade-off in cyber-physical systems. *IEEE Netw.* **30**(2), 62–66 (2016)
49. H Zhang, P Cheng, L Shi, J Chen, Optimal denial-of-service attack scheduling with energy constraint. *IEEE Trans. Autom. Control* **60**(11), 3023–3028 (2015)
50. O Landsiedel, E Ghadimi, S Duquennoy, M Johansson, in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*. Low power, low delay: opportunistic routing meets duty cycling (2012), pp. 185–196

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

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

---

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

---