Peer-to-Peer Netw. Appl. (2018) 11:1038–1059
DOI 10.1007/s12083-017-0565-5

Big program code dissemination scheme for emergency software-define wireless sensor networks

Xiao Liu[1] · Gaoxiang Li[1] · Shigeng Zhang[1] · Anfeng Liu[1]

Received: 9 January 2017 / Accepted: 21 April 2017 / Published online: 23 May 2017

Abstract Program codes as one of big data should be disseminated to all sensor nodes in a wireless software-define smart network (WSDSN) quickly. Due to the limited energy of sensor nodes, sensor nodes adopt asynchronous duty-cycle model to save energy. But neighbor nodes with sleep status can't receive program codes, resulting in longer transmission delay for spreading program codes. In this paper, an adjustable duty cycle based fast disseminate (ADCFD) scheme is proposed for minimum-transmission broadcast (MTB) in a smart wireless software-define network. In an ADCFD scheme, the duty cycle of nodes are adjusted to receive program codes timely. Thus, the transmission times and emergency transmission delay are reduced. The theoretical analysis and experimental results show that compare to previous broadcast scheme, the number of transmission in an ADCFD scheme is reduced by 44.776%–118.519%, the delay from disseminating program codes is reduced by 17.895%- 107.527%, while retaining network lifetime.

Keywords Big program code · minimum-transmission · broadcast · dissemination delay · lifetime

✉ Anfeng Liu
afengliu@mail.csu.edu.cn

Xiao Liu
xiaoliu@csu.edu.cn

Gaoxiang Li
1511872579@qq.com

Shigeng Zhang
sgzhang@csu.edu.cn

[1] School of Information Science and Engineering, Central South University, Changsha, China

1 Introduction

Due to various digital devices connect to cloud computing network as well fog computing network, data is generated in an explosively increasing manner. The development of big data imposes heavy burden to the underlying infrastructure. Because it requires vast computation, storage and communication resources. But the big data not only contains the sensed data, but also contains the process of diffusing program codes. One important application of big data is to diffuse program codes. In fog computing network, in order to enhance the intelligence of edge access devices, a large number of devices are updated, upgraded and reconfigured using wireless software defined network, so that the deployed devices obtain new vitality. Programming codes need spread to devices, the process is an inverse operation of big data collection. The process of program codes diffusion from data collection center to network edge is widely used in practical applications. For example: wireless sensor network (WSN) is a promising platform, and it is widely used in military and civilian domains [1–4]. In a smart wireless software-define network, many sensor nodes are deployed in the monitored area, sensor nodes sense data from surrounding environment, then the sensed data is transmitted to base station. The base station sends message to devices after the base station receives message, then the devices update and reconfigure software. Compare to traditional industrial automation systems using wired communications, a smart wireless software-define network brings notable advantages including lower cost, higher flexibility, and self-organizing capability, which significantly improves industrial efficiency and productivity [5].

With the rise of software defined network (SDN), the deployed devices can have new function by updating and configuring program codes. Due to the flexibility of software update and configuration, it has attracted the attention of researchers. In such environment, the program codes is usually transmitted to all devices. After the devices receive new program codes, the program codes in devices have been recompleted, then those devices have new functions. Broadcast is a fundamental operation in wireless software defined sensor networks (WSDSNs) [6, 7, 9]. Given a base station, the aim is to propagate data packets to all nodes in a minimum-transmission broadcast (MTB) while incurring minimum latency. This problem, called minimum-transmission broadcast (MTB), has been studied extensively [6–9]. Many applications such as industrial automation often have very stringent requirements on communication transmission delay [5]. Nevertheless, such environments faces severe challenges on the design of efficient codes diffusion in WSDSNs. First, energy of sensor nodes are limited, sensor nodes usually adopt duty-cycle style, in which a node alternates between dormant and active state [10–12]. Due to duty cycle style of nodes, program codes need transmit multiple time to nodes in the network. As a result, the MTB problem in duty-cycled networks (MTB-DC problem) is difficult. Second, the issue of tradeoff between lifetime and delay is a challenge task. Because duty cycle of node should be big to transmit program codes to all nodes in the network quickly, and duty cycle of nodes should be as small as possible to prolong network lifetime. Therefore, how to reduce the network transmission delay while retaining network lifetime is a challenge issue.

Though there are many researches about program code (or other large big data) broadcast in WSDSNs, it is still a place for further study about network lifetime, minimum transmission broadcast and broadcast delay. In this paper, an adjustable duty cycle based fast disseminate (ADCFD) approach is proposed for minimum-transmission broadcast (MTB) in the WSDSNs. The main contributions of the ADCFD approach are as follows:

(1) ADCFD approach optimizes broadcast, which can significantly improve broadcast performance while retaining original broadcast protocol. It has been widely used in WSDSNs. All previous schemes optimize scheduling algorithm under the condition of the given duty cycle, thus, the ability for improving network performance is limited. ADCFD scheme adopts different methods that increase duty cycle of nodes far from the sink to reduce the number of transmissions and broadcast delay. Because the number of broadcasts and delay has an inverse relationship with duty-cycle.

In this paper, we not only consider the energy consumption for transmitting program codes, but also consider the energy consumption of the whole network. Thus, the energy left can be full used to improve broadcast performance. According to [14], the energy consumption of nodes near to the sink is far higher than energy consumption of nodes far from the sink in the process of data collection. The data collection operation is regular operation in WSDSNs, updating and broadcasting program code is temporary. Therefore, the energy left of nodes far from the sink can be used to increase duty cycle of nodes far from the base station to improve broadcast performance.

(2) The optimization parameters and duty cycle are discussed and calculated for ADCFD scheme in theory. And the performance of the approach is given in this paper in theory. Through simulation experiment, broadcast times, broadcast delay can be reduced and energy-efficient can be enhanced in ADCFD scheme simultaneously. Compare to the previous scheme, the broadcast delay can be reduced by 17.895%–107.527%, the number of broadcast is reduced by 44.776%–118.519%. The ADCFD scheme has better performance while retaining network lifetime.

The rest of this paper is organized as follows: In section 2, the related works are reviewed. The system model is described in section 3. In section 4, a novel ADCFD approach is presented for WSDSNs. The performance analysis for ADCFD approach is provided in section 5. Section 6 are experimental results and comparison. We conclude in section 7.

2 Related work

In the wireless software defined network, in order to enhance the intelligence of access network devices, when software in network devices need to be updated, upgraded, and re-configured, the data control center (the sink) define software, and then transmit program codes with specific function to devices in the network, thus, a large number of software in devices can be updated, upgraded. Thus, the deployed device obtained new vitality. The broadcast in WSDSNs have been formulated and investigated in [13–19]. It can be divided into the following kinds based on the different applications requirements.

(1) The minimum-transmission broadcast (MTB) problem. In those researches, the aim of those schemes is how to reduce the number of times. In previous schemes, considering nodes are in active status all the time, thus, reducing the number of transmission is to find a Minimum Connected Dominating Set (MCDS) of the network [14, 15]. Nodes in the set can cover the whole network. Thus, as long as nodes in the set broadcast program codes once, all nodes in the network can receive program codes. In [17], the authors proved that building a minimum flooding tree is identical to finding an MCDS.

In fact, the aim of reducing transmission times is to reduce energy consumption. There are many researches about reducing energy consumption. Ref. [14] proposed a Heuristic broadcast algorithm BIP (broadcast incremental power). It builds a tree whose root is source node, in the process of constructing tree, one node uncovered is joined the broadcast tree, the energy cost in this way is minimum. Ref. [15] further optimize the BIP algorithm, proposed effective search algorithm r-shrink, it reschedules the non-leaf nodes in the process of building tree to reduce the total energy consumption of the radio.

In order to save energy in WSNs, a node alternates between dormant and active state [7]. In broadcast, the program codes need transmit many times to all nodes in the network. As a result, it becomes a problem in MTB-DC problem that how to select set of forwarding nodes and identify broadcast schedules.

Le Duc et al. [21] proposed Level-Based Approximation Scheme to identify forwarding nodes and their corresponding receivers for all time slots; then constructs a broadcast backbone by connecting these forwarding nodes to the broadcast source.

(2) The minimum latency broadcast scheduling (MLBS). In those schemes, they not only reduce energy consumption of nodes, but also reduce the time for transmitting program codes [13, 20].

Zhao D et al. [16] considered MLBS in duty cycled WSNs, and gives two algorithms, BS-1 and BS-2, the maximum broadcast delay in this two algorithms are $(\Delta - 1)TH$ and $13TH$, respectively. Here, Δ is the maximum degree of nodes, T denotes the number of time slots in a working cycle, and H is the broadcast latency lower bound obtained from the shortest path algorithm.

Khiati M et al. [22] proposes a Broadcast over Duty-Cycle and LEACH (BOD-LEACH) protocol, takes advantage of the LEACH's energy-efficient clustering. The new common static and dynamic broadcast periods are included to accelerate broadcasting. The dynamic periods are scheduled following the past arrivals of messages and using a Markov chain model.

3 System model and problem statements

3.1 The network model

We consider that n homogenous sensor nodes and one sink node deployed in a WSDSN. The sink node is regarded as the base stations, s means the base station. The network can be regarded as $\mathscr{M} = \{s, v_1, v_2, v_3, v_4 \ldots v_n\}$. v_i is the node ID. Those sensor nodes deployed in the target area such as environment monitor, industrial field, smart field and so on [23–25]. The network radius is R, the transmission radius of a sensor node is r. The sensor nodes are powered by battery whose energy are limited, and the energy of the sink is unlimited. Fig. 1 shows the initial connectivity setup of the network or the weighted graph established by a routing protocol. The network structure studied in this paper is given in the Fig. 1, s is the center of data collection. v_i is the sensor node in the network.

In consequence of saving energy, it would be better that nodes are set to sleep-wake when nodes don't need to receive program codes. Each node adopts asynchronous duty-cycle model, the duty cycle starts over between two consecutive periods. Each node has active/sleep two models. Each working cycle T is divided into |T| time slots with same length. Thus, one working cycles can be expressed as {0, 1, 2, 3,….., |T| − 1} time slots. It is shown as Fig. 5, considering working cycle of a node is 3, the working cycle is regarded as three time slots, they are 0, 1 and 2. Each node randomly selects active time slots. s is the sink node, the sink node sends program codes to neighbor nodes, then the program codes are transmitted from those neighbor nodes to outside nodes, until program codes reach the network boundary. The number in nodes are active time slots of nodes.

3.2 Energy consumption model

In this paper, nodes can periodically switch on and off according to a normalized duty cycle. The energy consumption of a node contains three aspects. (1) The energy consumption for receiving data packets, (2) the energy consumption for sending data packet, (3) When

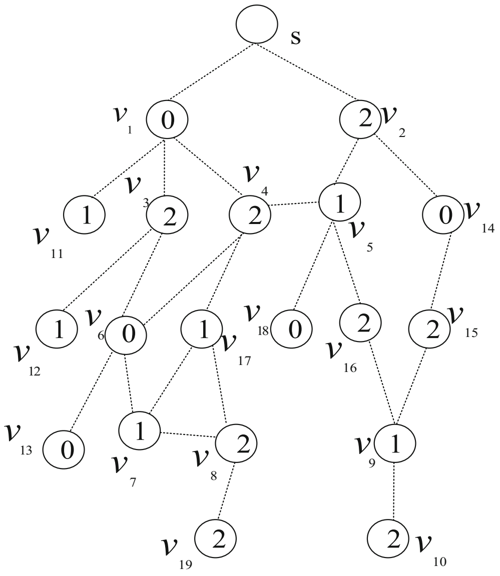

Fig. 1 Network structure diagram

node v_i at lm away from the sink is in the sleep state, the energy consumption of node v_i for sensing message is e^i_{sen}, the energy consumption for sensing data in different schemes are the same. When node v_i at lm away from the sink is in active status, the energy consumption of node v_i is e^i_r, and the energy consumption of node v_i at lm away from the sink is e^i_s when node v_i sends program codes to neighbor nodes. Thus, the total energy consumption e^i of node v_i at lm away from the sink is as follows.

$$e^i = e^i_r + e^i_{sen} + e^i_s \tag{1}$$

Thus, the power consumption of node contains (1) The power for receiving packets ϖ^i_r, (2) the power for sending packets ϖ^i_s. Thus the energy consumption at node v_i at lm away from the sink is as follows.

$$e^i = e^i_{sen} + \varpi^i_r Q^i_r + \varpi^i_s Q^i_s \tag{2}$$

Q^i_r is the amount of receiving data of node v_i at lm away from the sink, Q^i_s is the amount of sending data of node v_i at lm away from the sink.

The main parameters of system model adopted in this paper are similar to those parameters in [26–28]. The parameters are adopted in Table 1.

3.3 The problem statements

Definition 1: Minimization network updated delay. The network updated delay D is refer to the time for transmitting program codes from the base station to all nodes in the network. The aim of this paper is to reduce network updated delay, it can be calculated as follows.

$$\min(D) = \min\left(\textstyle\sum_{i\subseteq \mathrm{route}} d_i\right) \tag{3}$$

Table 1 Main notations and values adopted in this paper

Symbol	Description	Value
τ_a	Duty cycle	Calculation-specific
τ_c	Working cycle	250 ms
ϖ_r	The power used by a node to receive a packet;	Calculation-specific
ϖ_s	The power used to transmit an alert packet	Calculation-specific
ς_s	Transmission power consumption	0.0511 W
ς_r	Reception power consumption	0.0588 W
Θ_p	Preamble duration	0.26 ms
Θ_a	Ack window duration	0.26 ms
Θ_d	Packet duration	0.93 ms

Definition 2: Minimization broadcast times. Broadcast time ϱ is refer to the number of transmission when program codes are transmitted to the whole network. When nodes transmit program codes to neighbor nodes, the neighbor nodes can receive program codes only when nodes are in active statues. The aim of this paper is to reduce the number of transmission by increasing duty cycle of nodes far from the base station. The minimum broadcast times is expressed as follows.

$$\min(\varrho) = \min\left(\textstyle\sum_{i\subseteq \mathrm{route}} \varrho_i\right) \tag{4}$$

Definition 3: Maximization of network lifetime (denoted as ℓ). The fundamental purpose of application is to maximize network lifetime [29–31]. Network lifetime can be defined as the time of the node that dies first. For after the first node dies, it may affect the connectivity and coverage of the network severely leading the network cannot play a proper role. Thus, network lifetime in this paper is the time of the node that dies first. The energy consumption of node e^i contains the energy consumption e^i_r for receiving data packet, the energy consumption e^i_s for sending data packet and the energy consumption for sensing data e^i_{sen}. Where E_{init} represents the initial energy of node v_i. Thus, network lifetime is as follows.

$$\max(\ell) = \max_{0\le i\le n}\left(E_{\mathrm{init}} \Big/ \left(e^i_r + e^i_{sen} + e^i_s\right)\right) \tag{5}$$

Definition 4: Maximization of effective energy utilization rate (denoted as φ). φ refers to the ratio of energy efficiently utilized to the total energy of nodes in the network. The aim in this paper is to make full use of energy left in nodes far from the sink, then maximize energy utilization of the whole network. e^i is the energy consumption of node v_i, E_{init} represents the initial energy of node v_i. Thus, the effective energy utilization rate is as follows.

$$\max(\varphi) = \max\left(\left(\textstyle\sum_{1\le i\le n} e^i\right) \Big/ \left(\textstyle\sum_{1\le i\le n} E_{\mathrm{init}}\right)\right) \tag{6}$$

Obviously, the goal of ADCFD is to minimize delay, D, minimize broadcast time, ϱ, maximize the network life, ℓ, and effective energy utilization, φ, which can be summarized as follows.

$$\begin{cases} min(D) = min\left(\sum_{i\subseteq \mathrm{route}} d_i\right) \\ min(\varrho) = min\left(\sum_{i\subseteq \mathrm{route}} \varrho_i\right) \\ max(\ell) = \max_{0\le i\le n}\left(E_{\mathrm{init}} \Big/ \left(e^i_r + e^i_{sen} + e^i_s\right)\right) \\ max(\varphi) = max\left(\left(\sum_{1\le i\le n} e^i\right) \Big/ \left(\sum_{1\le i\le n} E_{\mathrm{init}}\right)\right) \end{cases} \tag{7}$$

4 Main design of ADCFD

4.1 Research motivation

The research motivation in this paper is based on the following two aspects:

(1) When a sensor node in a wireless software-define network senses surrounding environment, the sensed data can be transmitted to the sink. Due to nodes near to the sink can bear a large amount of data, its energy consumption is higher than the energy consumption of nodes far from the sink. The energy consumption is given in Fig. 2. According to some researches, there are 90% energy left when the network is dead [31]. That is to say, the energy left in the network is wasted. The paper mainly researches program code transmission. Program codes are transmitted to all network, then software in devices can be updated. The aim in this paper is to reduce transmission delay. Due to the much energy left in a network, the energy left can be used to increase duty cycle of nodes, which can reduce the transmission delay.

(2) The higher number of transmission and broadcast delay in the previous scheme.

LBAS scheme proposed in Ref. [21] constructs a broadcast backbone for reducing broadcast delay and transmission times. Before program codes start transmitting, the broadcast backbone should be built. The broadcast backbone construction is accomplished by a two-stage traversal on all forwarding nodes. The process of building broadcast backbone is as follows.

For each time slot, a covering set with a minimum number of nodes is identified such that these nodes can cover all nodes active at that time slot. Nodes in such a set are called covering nodes, and nodes covered by these covering nodes are called covered nodes.

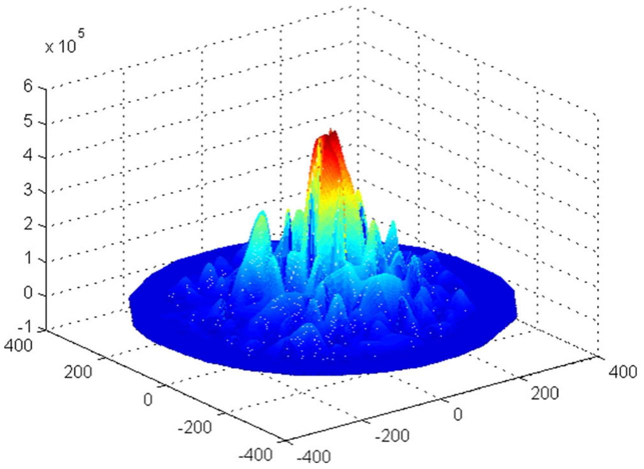

Fig. 2 The energy consumption for collecting data in wireless software-define networks

(1) Finding the minimum covering node set

Considering the broadcast source s is a covering node, but not as a covered node in any case. σ_a is the set of nodes with active time slot a. ω_a is the minimum covering node set of nodes with active time clot a [21]. For each time slot $a \in$ T, if the number of neighbor nodes with active time slot a of node y is largest in σ_a, node y can be covering node. This process is executed until all nodes in σ_a are covered.

For example, considering the network with 20 nodes in the Fig. 3, the duty-cycle |T| is 3, there are 0, 1 and 2 time slots [21]. For the time slot 0, the neighbor nodes of v_6 are v_3, v_4, v_7, v_{12}, v_{13}, there are two nodes whose active time are 0. There are two nodes whose active time are 0 in those neighbor nodes of node v_5. There are one nodes whose active time are 0 in those neighbor nodes of node v_2. The minimum covering node set for time slot 0 is ω_0={v_2 , v_5 , v_6}. Because neighbor nodes of three nodes (v_2 , v_5 , v_6) are include all nodes whose active time is 0. The minimum covering node set 1 is ω_1={v_1 , v_3 , v_7 , v_{10}}, the minimum covering node set 2 is ω_2={v_4 , v_8 , v_9}.

(2) Constructing broadcast backbone

The broadcast backbone is built based on the minimum covering node sets. It is can be seen that if all nodes in the minimum covering node sets are connected as a tree in the network, the program codes can be transmitted to all nodes in the network along the broadcast backbone. Because nodes in the minimum covering node sets cover all nodes in the

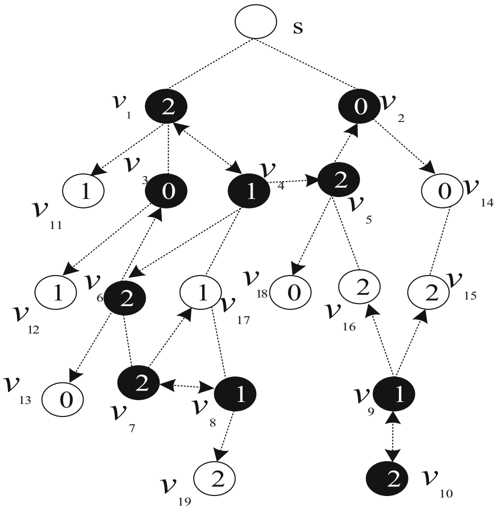

Fig. 3 Network structure diagram in wireless software-define networks

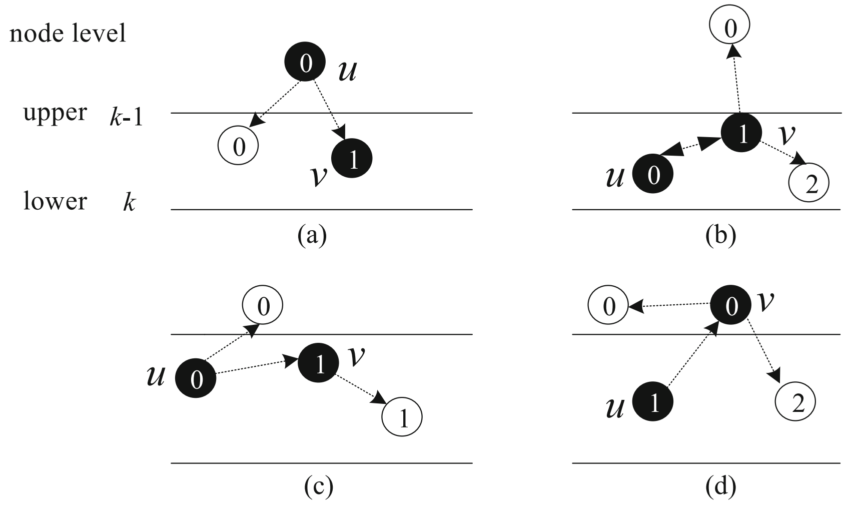

Fig. 4 The way for connecting covering nodes in wireless software-define networks

network. Thus, connecting all nodes in the minimum covering node sets to build broadcast backbone to reduce transmission. The principle of this paper is as follows.

In the LBAS scheme, nodes are divided into several levels, nodes near to the sink have upper level, and nodes far from the sink have lower level. As shown in Fig. 4.

(3) If a covering node near to the sink has a neighbor node in the sets of covering node, and the covering node covers the neighbor node, the node will connect to his neighbor node. An example is shown in Fig. 4a, *u* is the parent of node *v*.

(4) If a covering node covers the neighbor node in the covering sets, and the neighbor node covers the covering node [21], the two nodes are in the same level near to the sink. One can connect to other node. An example is shown in Fig. 4b.

(5) If one covering node covers the neighbor node in the covering sets, and the covering node and the neighbor node are in the same level [21], the covering node is selected as the parent. An example is shown in Fig. 4c, where node *u* is selected as the parent of node *v*.

Though some nodes are selected for building broadcast backbone, some nodes may not be selected. Such as Fig. 4d, node *v* forms a covering sub-tree with itself.

Through the following step, a broadcast backbone can be completed. When the source node (the sink) sends program codes to neighbor nodes, and program codes are transmitted along the broadcast backbone. At last, all nodes in the network can receive program codes.

Though LBAS scheme can reduce the number of transmission and broadcast delay to a certain extent, the performance of LBAS scheme is improved only from adjusting the algorithm, the ability of improving performance is limited. The duty cycle in the LBAS scheme is showed in Fig. 5. The working cycle in LBAS scheme is T, each working cycle is divided into |T| time slots. Each sensor node selects a time slot as an active time slot, sensor node in an active time slot can receive data packets. It is shown in Fig. 5, the length of working cycle T is 3, and it contains 0, 1 and 2 time slots. Only when one node is in an active time slot, the node can receive data packets. If node A want to transmits data, the next node B is in sleep time slot, node A must wait for a while, until node B is in active time slot. This process can cause longer transmission delay.

Though program codes can be transmitted to all nodes in the network, the transmission delay is still longer. The reason is that when one node receives the program codes from the sink, the program codes is only transmitted to the nodes when nodes is in the active time slot. Each node randomly selects a time slots as an active time slot, if the active time slots of nodes in the lower level in the broadcast backbone is higher than the active time slots of nodes in the upper level, the program codes are transmitted to the lower level nodes until the next working cycle, it needs longer transmission delay. As

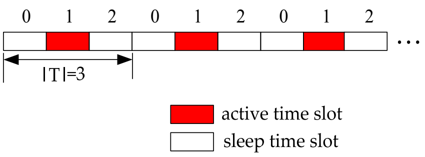

Fig. 5 the time slots of the LBAS scheme in wireless software-define networks

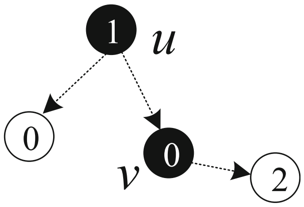

Fig. 6 the level in wireless software-define networks

showed in the Fig. 6, the active time slot of node u in the upper level is 1, the active time slot of node v in the lower level is 0, node u wait for the next working cycle to transmit the program codes to node v when node u receives program codes. It need longer transmission delay.

It can be seen that duty cycle of nodes in the LBAS scheme in different area are the same. Each node selects one time slots as an active time slot, which can cause larger transmission delay and higher energy left in the no-hotspots area.

Due to reduce broadcast delay farther, ADCFD scheme adjusts duty cycle of nodes to reduce broadcast delay. In the ADCFD scheme, duty cycle of nodes are adjusted based on energy left of the nodes. The duty cycle of a node in the ADCFD is showed in the Fig. 7. Considering the working cycle T is 3, it is divided into 0, 1 and 2 time slots. Nodes with higher energy consumption can select two or more time slots as active time slots. The different of duty cycle of nodes in the ADCFD scheme from LBAS scheme is that nodes with higher energy left can select many time slots as active time slots. In the above analysis, each node sends the detected data to the sink. Therefore, the energy consumption is imbalanced in the whole network. Nodes far from the sink have much energy left, the duty cycle of those nodes can be increased to reduce transmission delay. It is showed that the longer the duty cycle is, the transmission delay is lower.

Broadcast performance depends on duty cycle of nodes in the network. Larger duty cycle brings higher monitoring performance. Figure 8 shows the number of broadcast in different duty cycle. The number of broadcast decrease with the increase of duty cycle of nodes. Figure 9 shows the broadcast delay in different duty cycle. It can be seen that the number of broadcast and transmission delay decrease with the increase of duty cycle of nodes. The larger duty cycle of nodes are, and the probability of nodes that receive program codes is higher when one node broadcasts program codes once. Thus, the number of broadcast and transmission delay are smaller.

Figure 10 shows that the remaining energy of nodes is used for increasing duty cycle in different regions. It can be seen from Fig. 2 that duty cycle of non-hotspot nodes are higher than duty cycle of hotspot nodes. According to the above analysis, there are much energy left in the non-hotspots area, the energy left can be used to increase duty cycle of nodes. The adjustment duty cycle of nodes in different areas is given in Fig. 10.

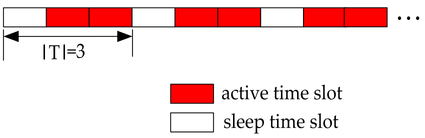

Fig. 7 the time slots of the ADCFD scheme in wireless software-define networks

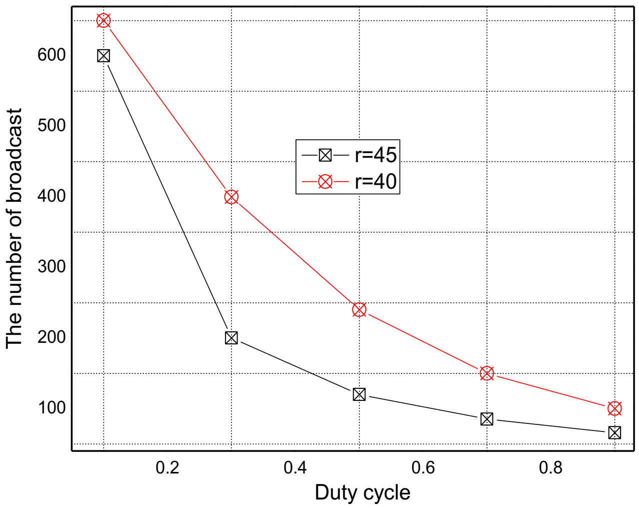

Fig. 8 The number of broadcast with different duty cycle

In the previous scheme, the duty cycle of nodes in different areas are the same. But the duty cycle of nodes far from the sink in the ADCFD scheme are greater than that of previous scheme. Adjusting the nodal duty cycle dynamically can achieve better and more comprehensive broadcast performances. For this reason, an adjustable duty cycle based fast disseminate (ADCFD) approach is proposed to improve program codes diffusion while retaining network lifetime.

4.2 The calculation of energy consumption of the network

This section gives the amount of data loaded in different area, then the energy consumption in different area for collecting data is shown.

Inference 1. Considering the probability of producing packet of one node is λ, the node v_i at l, $l = hr + x$ $(0 < x < r)$ away from the sink receives the amount of data

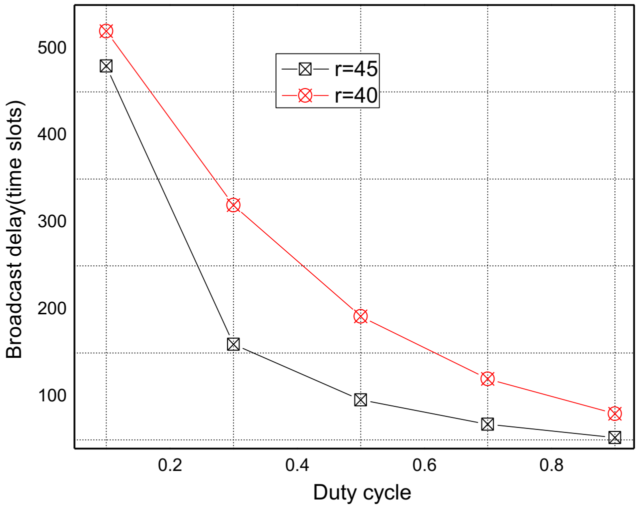

Fig. 9 Broadcast delay with different duty cycle

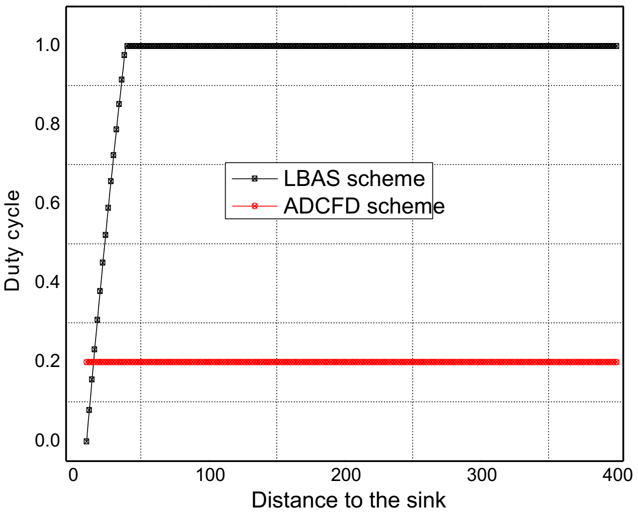

Fig. 10 duty cycle for different distances

Q_r^i, it is as follows.

$$Q_r^i = \left(z + \frac{z(z+1)r}{2l}\right)\lambda \tag{8}$$

Proof: It can be seen from Eq. (8), one node generates a data packet in a collection cycle. If the probability is λ. If one node produces a data packet, the number of data packet at node v_i is $z + \frac{z(z+1)r}{2l}$, thus, the amount of data loaded by the nodes at l away from the sink is as follows.

$$Q_r^i = \left(z + \frac{z(z+1)r}{2l}\right)\lambda$$

Inference 2. Considering the probability of generating packet of one node is λ, the node v_i at l, $l = hr + x$ (0 < x < r) away from the sink sends the amount of data Q_s^i, it is as follows.

$$Q_s^i = \left((z+1) + \frac{z(z+1)r}{2l}\right)\lambda \tag{9}$$

Proof: It can be seen from Eq. (8), when one node receives the data packet, it will generate one data packet with probability λ by itself, then those data packets are transmitted to the next node, the amount of data transmitted by the nodes at l away from the sink are as follows.

$$Q_s^i = Q_r^i\lambda = \left((z+1) + \frac{z(z+1)r}{2l}\right)\lambda$$

The amount of data of nodes varies from area to area, hotspot nodes have to receive and forward large amount of data packets. Considering nodes at l m away from sink have Q_z^i packets to be transmitted in a communication cycle τ_c, and Q_r^i packets to be received.

e^i denotes the total energy consumption of node v_i at l m away from sink in data working cycle τ_c. According to Refs. [32, 33], there are three states in the energy consumption e^i: (i) transmission, (ii) reception, and (iii) sensing. e^i can be computed as follows.

$$e^i = e_{sen}^i + \varpi_r^i Q_r^i + \varpi_s^i Q_s^i \tag{10}$$

ϖ_r^i is the power used for receiving a packet. ϖ_s^i is the power for transmitting a packet. It can be calculated as follows.

$$\varpi_s^i = \varsigma_s\Theta_d + \left(\frac{\tau_a\tau_c}{4(\Theta_p+\Theta_a)} + \frac{1}{2}\right)(\varsigma_s\Theta_p + \varsigma_r\Theta_a) \tag{11}$$

$\varsigma_s\Theta_d$ is the energy consumption for sending a data packet. $\varsigma_s$is the transmission power consumption, ς_r is the reception power consumption, Θ_d is the packet duration, Θ_a is the ACK window duration, Θ_p is the preamble duration. τ_a is the active period radio, τ_c is the working cycle.

ϖ_r^i can be calculated as follows.

$$\varpi_r^i = \varsigma_r\Theta_d + (\varsigma_r\Theta_p + \varsigma_s\Theta_a) \tag{12}$$

$\varsigma_r\Theta_d$ is the energy consumption for receiving a packet and $\varsigma_r\Theta_p + \varsigma_s\Theta_a$ is for the reception of the preamble and the transmission of the returned message. Eq. (2) calculates the total energy consumption.

The energy consumption for receiving and sending data packet in previous scheme are given in Fig. 11 and Fig. 12, respectively. In the previous scheme, the duty cycle of all nodes are the same. The energy consumption for receiving and sending data packet is higher, and the energy consumption of nodes is

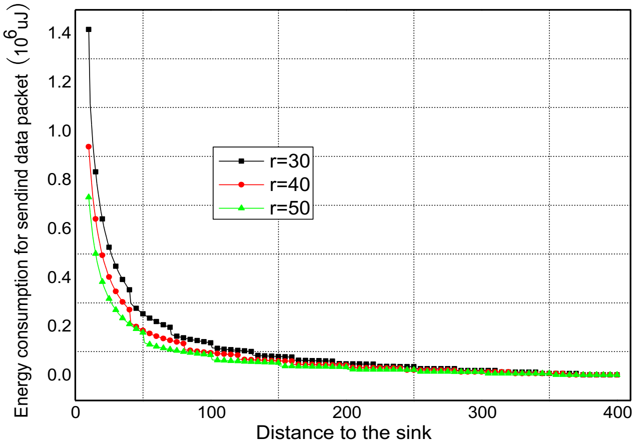

Fig. 11 Energy consumption of receiving data

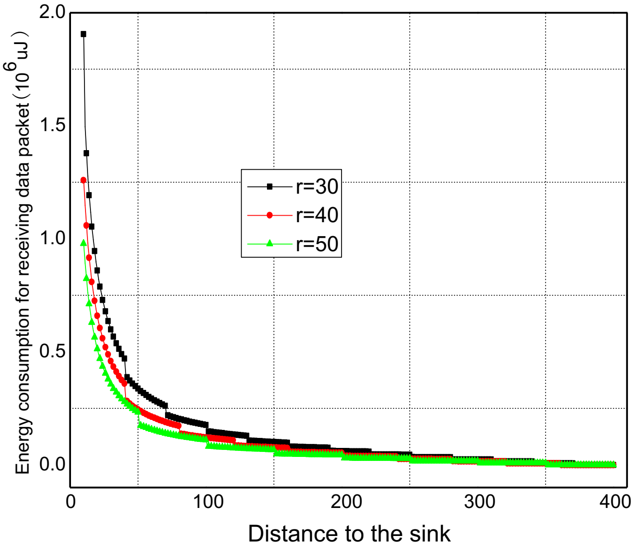

Fig. 12 Energy consumption of sending data

unbalanced. The energy consumption for receiving and sending data packet in the previous scheme are given in Fig. 13. It can be seen that the energy consumption of nodes near to the sink is higher than the energy consumption of node far from the sink. There are enough energy left in the non-hotspots to increase duty cycle to reduce the number of transmission and transmission delay.

Theorem 1. Considering the initial energy of each node is E_{init}. The remaining energy E_{left}^i of data operation of node v_i at l m away from the sink in a data collection period τ_c is as follows.

$$E_{\text{left}}^i = E_{\text{init}} - e_{sen}^i - \varpi_r^i \mathrm{Q}_r^i - \varpi_s^i \mathrm{Q}_s^i \tag{13}$$

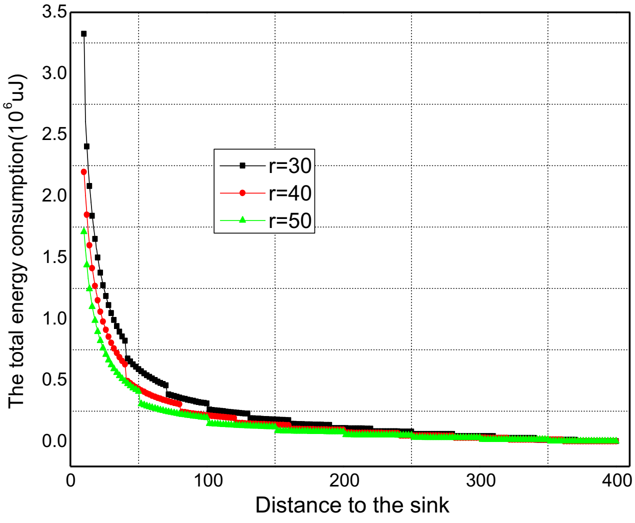

Fig. 13 The total energy consumption

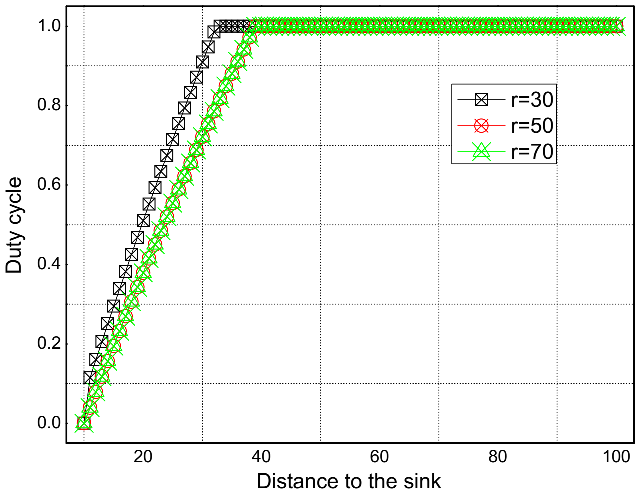

Fig. 14 The adjustment duty cycle in the ADCFD scheme

Proof. The energy consumption of node i at l m away from the sink is $e_r^i + \varpi_r^i Q_r^i + \varpi_s^i Q_z^i$, the initial energy of each node is E_{init}. Thus the remaining energy of data operation of node v_i is as follows.

$$E_{\text{le ft}}^i = E_{\text{init}} - e_{sen}^i - \varpi_r^i Q_r^i - \varpi_s^i Q_s^i$$

4.3 Adjustable duty cycle calculation

The main difference from ADCFD scheme to previous scheme is the different duty cycles of nodes. Therefore, the calculating method of sensing and communicating duty cycle in different regions of the network is given in this section.

Theorem 2. Considering the number of sending and receiving packets in hotspots are Q_s^{max} and Q_r^{max}, respectively. The residual energy is only used for increasing τ_a, the active

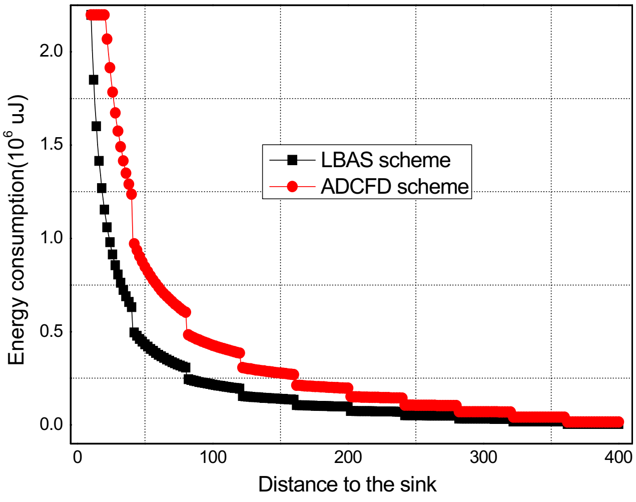

Fig. 15 The energy consumption in two schemes

period of node v_i at l m away from sink is τ_a^i, and the active period of nodes nearest to the sink is τ_a^{hotspots}. τ_a^i is calculated as follows.

$$\tau_a^i = \frac{\tau_a^{\text{hotspots}} Q_s^{max}}{Q_s^i} - 2\vartheta \frac{2\alpha\left(Q_r^i - Q_r^{max}\right) - 2\varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right) + \beta\left(Q_s^i - Q_s^{max}\right)}{\beta\tau_c}$$

Where $\alpha = \varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right), \beta$

$$= \varsigma_s\Theta_p + \varsigma_r\Theta_a, \vartheta = \left(\Theta_p + \Theta_a\right)\frac{\tau_c}{Q_s^i} \tag{14}$$

Proof. The scheme proposed in this paper increases active period to reduce the number of transmission and transmission delay while retaining network lifetime. Nodes in hotspots area consume largest energy according to Refs. [34, 35], the number of sending and receiving packets in hotspots are Q_s^{max} and Q_r^{max}, respectively. In order to ensure the network lifetime, the maximization energy consumption of nodes far from the sink can equal to the energy consumption of nodes in hotspots area. It is as follows.

$$\varpi_s^i = \varsigma_s\Theta_d + \left(\frac{\tau_a\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right)\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right), \varpi_r^i$$
$$= \varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)$$

$$e_{sen}^i + \varpi_r^i Q_r^i + \varpi_s^i Q_s^i = e_{sen}^i + \varpi_r^{max} Q_r^{max} + \varpi_s^{max} Q_s^{max}$$

$$\Rightarrow \varpi_r^i Q_r^i - \varpi_r^{max} Q_r^{max} = \varpi_s^{max} Q_s^{max} - \varpi_s^i Q_s^i$$

$$\Rightarrow \left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right) Q_r^i - \left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)$$
$$Q_r^{max} = \left(\varsigma_s\Theta_d + \left(\frac{\tau_a^{\text{hotspots}}\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right)\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)\right)$$
$$Q_s^{max} - \left(\varsigma_s\Theta_d + \left(\frac{\tau_a^i\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right)\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)\right) Q_s^i$$

$$\Rightarrow \left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right)$$
$$= \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right) + \left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)$$
$$\left(\left(\frac{\tau_a^{\text{hotspots}}\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right) Q_s^{max} - \left(\frac{\tau_a^i\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right) Q_s^i\right)$$

$$\Rightarrow \frac{\left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right) - \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right)}{\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)}$$
$$= \left(\left(\frac{\tau_a^{\text{hotspots}}\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right) Q_s^{max} - \left(\frac{\tau_a^i\tau_c}{4\left(\Theta_p + \Theta_a\right)} + \frac{1}{2}\right) Q_s^i\right)$$

$$\Rightarrow \frac{\left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right) - \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right)}{\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)}$$
$$* \left(4\left(\Theta_p + \Theta_a\right)\right)$$
$$= \left(\left(\tau_a^{\text{hotspots}}\tau_c + 2\Theta_p + 2\Theta_a\right) Q_s^{max} - \left(\tau_a^i\tau_c + 2\Theta_p + 2\Theta_a\right) Q_s^i\right)$$

$$\Rightarrow \frac{\left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right) - \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right)}{\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)\tau_c}$$
$$* \left(4\left(\Theta_p + \Theta_a\right)\right) + \left(2\Theta_p + 2\Theta_a\right)\left(Q_s^i - Q_s^{max}\right)$$
$$= \left(\tau_a^{\text{hotspots}}\tau_c Q_s^{max} - \tau_a^i\tau_c Q_s^i\right)$$

$$\Rightarrow (\frac{\left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right) - \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right)}{\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)\tau_c}$$
$$* \left(4\left(\Theta_p + \Theta_a\right)\right) + \left(2\Theta_p + 2\Theta_a\right)\left(Q_s^i - Q_s^{max}\right))\tau_c$$
$$= \left(\tau_a^{\text{hotspots}} Q_s^{max} - \tau_a^i Q_s^i\right)$$

$$\Rightarrow \tau_a^{\text{hotspots}} Q_s^{max} - (\frac{\left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right) - \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right)}{\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)\tau_c}$$
$$* \left(4\left(\Theta_p + \Theta_a\right)\right) + \left(2\Theta_p + 2\Theta_a\right)\left(Q_s^i - Q_s^{max}\right))\tau_c = \tau_a^i Q_s^i$$

$$\Rightarrow \tau_a^i = \frac{\tau_a^{\text{hotspots}} Q_s^{max}}{Q_s^i} -$$
$$(\frac{\left(\varsigma_r\Theta_d + \left(\varsigma_r\Theta_p + \varsigma_s\Theta_a\right)\right)\left(Q_r^i - Q_r^{max}\right) - \varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right)}{\left(\varsigma_s\Theta_p + \varsigma_r\Theta_a\right)\tau_c}$$
$$* \left(4\left(\Theta_p + \Theta_a\right)\right) + \left(2\Theta_p + 2\Theta_a\right)\left(Q_s^i - Q_s^{max}\right))\frac{\tau_c}{Q_s^i}$$

$$\Rightarrow \tau_a^i = \frac{\tau_a^{\text{hotspots}} Q_s^{max}}{Q_s^i}$$
$$-2\vartheta \frac{2\alpha\left(Q_r^i - Q_r^{max}\right) - 2\varsigma_s\Theta_d\left(Q_s^{max} - Q_s^i\right) + \beta\left(Q_s^i - Q_s^{max}\right)}{\beta\tau_c}$$

Where $\alpha = \varsigma_r\Theta_d + (\varsigma_r\Theta_p + \varsigma_s\Theta_a)$, $\beta = \varsigma_s\Theta_p + \varsigma_r\Theta_a$, $\vartheta = \left(\Theta_p + \Theta_a\right)\frac{\tau_c}{Q_s^i}$.

Figure 14 shows the adjustment duty cycle of nodes in the ADCFD scheme. In the ADCFD scheme, the duty cycle of nodes far from the sink is increased depended on the energy left of nodes. It can be seen that the duty cycle of nodes far from the sink is higher, it up to 1, and the duty cycle of nodes near to the sink is lower.

4.4 Scheduling method of ADCFD scheme

The duty cycle of nodes are adaptively adjusted by the ADCFD scheme which use feedback control method, the aim of this paper is to reduce transmission delay while retaining network lifetime [36].

In the process of data collection, when one node v_i sends a data packet, the maximum energy consumption e^i_{max}, added in the header of the data packet, is set 0. Each node v_k receives a data packet, then compare the energy consumption e_k to the maximum energy e^i_{max} in the header of data packe, if $e^i_{max} < e_k$, then.

$$e^i_{max} = e_k$$

When the sink receives the data packet, the maximum energy consumption is also received by the sink. After all nodes send a data packet in one data collection cycle, the sink calculates the maximum energy consumption e_{max}.

In the process of program codes diffusion, when the sink sends program codes, the maximum energy consumption e_{max} is added in the header of program codes. Each node v_k receives program codes, then compare the energy consumption e_k to the maximum energy e^i_{max} in the header of data packet, if $e_{max} < e_k$, then.

$$e_{max} = e_k$$

The program codes are sent to all nodes in the network. After all nodes in the network receive program codes, the ACK message contains maximum energy consumption e_{max} is returned to the sink. Then the sink sends adjustment message to adjust duty cycle of nodes according to the energy consumption of nodes and the maximum energy consumption.

In the process of adjusting duty cycle, the sink sends adjustment message contains maximum energy consumption e_{max} to nodes for adjusting duty cycle of nodes. When one node receives the adjustment message, the duty cycle of the node is adjusted according to the Eq. (15) based on the energy consumption of the node and the maximum energy consumption. The process continues until reaches balanced status.

When sensor nodes send data to the sink, the message about energy is also transmitted to the sink, the energy consumption of nodes near to the sink can be obtained. If the energy consumption of nodes far from the sink is less than the energy consumption of nodes near to the sink, the duty cycle of nodes far from the sink can be adjusted using feedback message. But if the energy consumption of nodes far from the sink is bigger than the energy consumption of nodes near to the sink, the duty cycle of nodes far from the sink can be decreased using feedback message. The duty cycle of nodes can be adjusted better through several feedback control.

Algorithm 1: The ADCFD scheme for adjusting duty cycle

```
Initialize : Considering the active time slots of node $v_i$ are $\tau^i_a$, the energy consumption of node $v_i$ are $e_i$, the energy consumption $e_i$ is set 0.
Input energy consumption of each node
  Case 1: The process of data collection
  For each node $v_i$ Do
    Sends a data packet to the sink,
    Store the energy consumption $e^i_{max} = e_i$ in the data packet.
    For each node $v_k$ receives the data packet Do
      Calculates the energy consumption $e_k$ of the node $v_k$ according to the Eq.(11).
      If $e_k > e^i_{max}$ then
        $e^i_{max} = e_k$
      End if
    End for
    The sink stores the maximum the energy consumption $e^i_{max}$ for this data collection.
  End for
  The sink calculates the maximum energy consumption $e_{max}$ in the process of data collection.
  Case 2: The process of program codes diffusion
  The sink sends program codes to the neighbor nodes, and adds the maximum energy co nsumption $e_{max}$ in the header of program codes.
  For each node $v_k$ receives the program codes Do
    Calculates the total energy consumption $e_k$.
    Compare its own energy consumption, $e_k$, with the energy consumption, $e_{max}$. Stored in the header of the program codes.
    If $e_{max} < e_k$ then
      $e_{max} = e_k$
    End if
  End for
  The ACK message contains the maximum energy consumption $e_{max}$ will be returned to the sink
  Case 3: Adjust the duty cycle.
    The sink send adjustment message for a d-
    just the duty cycle.
  **For**each node $v_k$ receives the program codes **Do**
      Compare the energy consumption $e_k$ and the
          maximum energy consumption $e_{max}$,
      $e_k^{\Delta} = e_{max} - e_k$
      **If**$e_k^{\Delta} > 0$ then
          Compute $\tau_a^i$ according to the Eq.(15);
      **Else if**$e_k^{\Delta} > 0$ then
          Maintain the current duty cycle.
      **End if**
  **End for**
**Output** duty cycle of each node
```

In the above analysis, it can be known that nodes with maximum energy consumption are in the area near to the sink, the area called hotspots area. For the given duty cycle of nodes in the hotspots area, the duty cycle of nodes can be adjusted according to algorithm 1. At the first, if duty cycle of nodes in the hotspots area are big, the transmission delay and network lifetime can't be satisfied the required value. In order to prolong network lifetime, Algorithm 2 is given. It can be seen that in order to prolong network lifetime, the maximum energy consumption must be reduced, the method is to reduce the duty cycle of nodes in the hotspots area is given in the algorithm 2. The duty cycle can be reduced $\Delta\tau$, that is as follows.

$$\tau_{min} = \tau_{min} - \Delta\tau$$

Then adjusting duty cycle of other nodes in the network, until reaches balanced status.

Algorithm 2: The ADCFD scheme for prolong the network

```
**Initialize :** Considering the active time slots of node $v_i$ are $\tau_a^i$, the energy consumption of node $v_i$ are $e_i$, the energy consumption $e_i$ is set 0.
  Running the Algorithm 1, until the system reach balanced states.
//The system is balanced under the condition that the determined duty cycle $\tau_{hotspots}$ of nodes near to the sink. The duty cycle of other nodes is adjust according to the maximum energy consumption. The network lifetime is determined based on the maximum energy consumption of nodes in the network.
//prolong the network lifetime is to reduce the duty cycle of nodes with the maximum energy consumption.
  **When** the transmission delay is less than the required transmission delay **Do**
      Reduce the duty cycle $\tau_{min}$ of the node with the maximum energy consumption
      $\tau_{min} = \tau_{min} - \Delta\tau$.
      Compare the duty cycle of other nodes according to the Algorithm 1.
  **End for**
Output the duty cycle of each node.
```

5 Performance analysis in theory

5.1 Analysis of transmission times

Theorem 3. The number of nodes with k active time slot is N_k. Considering node v_i at l m away from sink, Λ_k^i is the set of neighbor nodes with k active time slot of node v_i. $|\Lambda_k^i|$ is the number of nodes in the set Λ_k^i. $\Upsilon_k = \{v_{i_1}, v_{i_2}, v_{i_3}, \ldots v_{i_n}\}$ is the set of nodes, the $|\Lambda_k^{i_1}| > |\Lambda_k^{i_2}| > |\Lambda_k^{i_3}| > |\Lambda_k^{i_4}| > \ldots \Lambda_k^{i_n}$ and

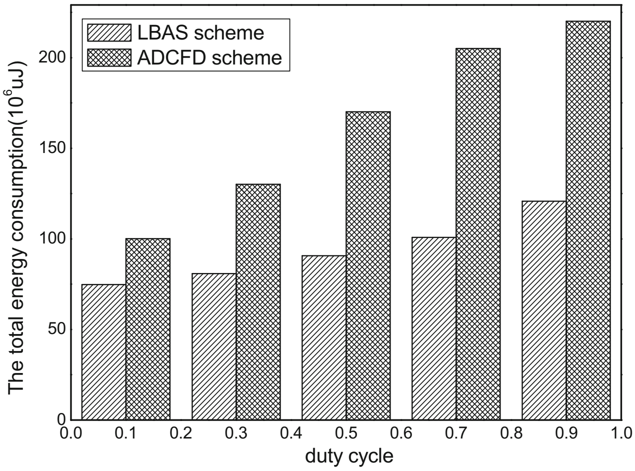

Fig. 16 The total energy consumption under different duty cycle in two schemes

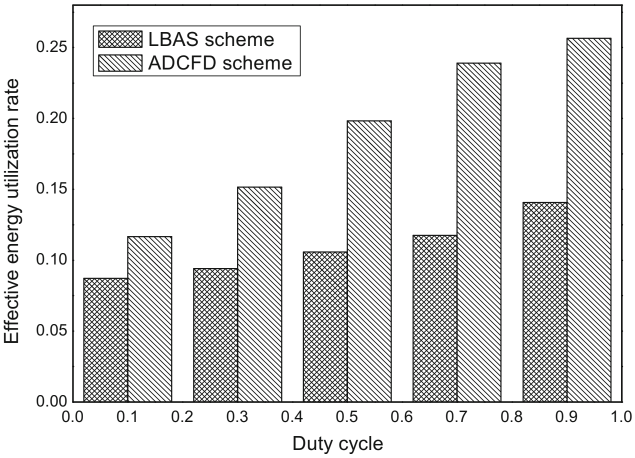

Fig. 18 The effective energy utilization rate under different duty cycle in two schemes

$|\Gamma_k| = N_k$, where the $\Gamma_k = \{\Lambda_k^{i_1}, \Lambda_k^{i_2}, \Lambda_k^{i_3}, \ldots \Lambda_k^{i_n}\}$. Considering the number of transmission in the network is Ξ, it can be calculated as follows.

$$\Xi = \sum_{k=0}^{|T|-1} |\Upsilon_k| \tag{15}$$

Proof. It can be seen that in order to reduce the number of transmission, the broadcast backbone is built. The first step is to find the minimum covering node sets. The method is that u_k is the set of nodes with active time slot k. c_k is the minimum covering node set. For each time slot $k \in \mathrm{T}$, one node y is selected as a covering code for time slot k if y covers the largest number of uncovered nodes in u_k. This process of selection is executed, until all nodes in u_k are covered. That is to say, for active time slot k, the minimum covering node with active time slot k is Υ_k. The total number of covering node is $\Omega = \{\Upsilon_1, \Upsilon_2, \Upsilon_3 \ldots \Upsilon_{|T|}\}$.

In the ADCFD scheme, the broadcast backbone is built based on the minimum covering node sets. For one node in the Ω, the node can be selected as a transmission node to transmit data. If all nodes in Ω transmit program codes once, all nodes in the network will receive program codes. Thus, the number of transmission is the number of nodes in the Ω. Thus, the number of transmission times is as follows.

$$\Xi = \sum_{k=0}^{|T|-1} |\mathrm{Y}_k|$$

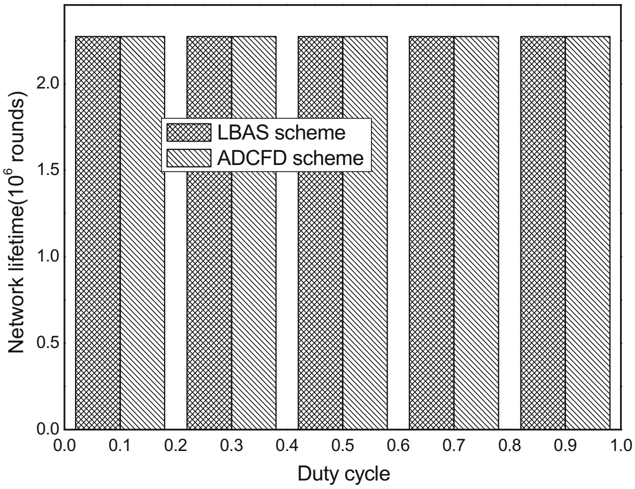

Fig. 17 The network lifetime under different duty cycle in two schemes

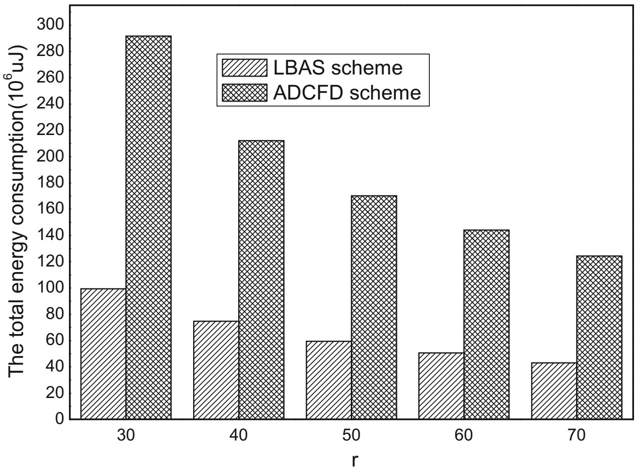

Fig. 19 The total energy consumption under different r in two schemes

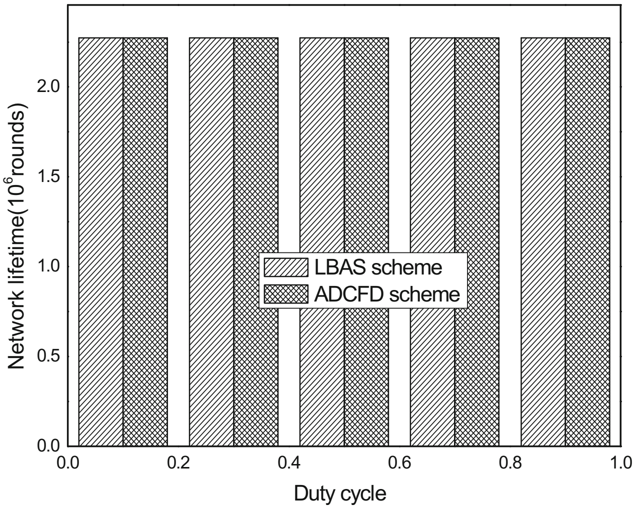

Fig. 20 The network lifetime under different r in two schemes

5.2 Analysis of broadcast delay

Theorem 4. The number of nodes with k active time slot is N_k. Considering node v_i at l m away from sink, Λ_k^i is the set of neighbor nodes with k active time slot of node v_i. τ_ϵ^i is the maximum active time slots of nodes in the set Λ_k^i, For example, if the active time slots of node A is 0, and the active time slots of node B is 1, the τ_ϵ^i is 1. $|\Lambda_k^i|$ is the number of nodes in the set Λ_k^i. $\Upsilon_k = \{v_{i_1}, v_{i_2}, v_{i_3}, \ldots v_{i_n}\}$ is the set of nodes, and the $|\Lambda_k^{i_1}| < |\Lambda_k^{i_2}| < |\Lambda_k^{i_3}| < |\Lambda_k^{i_4}| < \ldots \Lambda_k^{i_n}$ and $|\Gamma_k| = N_k$, where the $\Gamma_k = \{\Lambda_k^{i_1}, \Lambda_k^{i_2}, \Lambda_k^{i_3}, \ldots \Lambda_k^{i_n}\}$. $\Omega = \{\Upsilon_1, \Upsilon_2, \Upsilon_3 \ldots \Upsilon_{\mathrm{T}}\}$, considering the transmission delay is $\gimel$, it can be calculated as follows.

$$\gimel = \sum_{i \subseteq \Omega} \mathfrak{H}_i \qquad \begin{cases} \mathfrak{H}_i = \tau_a^i - \tau_{\beta_1} & \text{if} \quad \tau_a^i < \tau_\epsilon^i \\ \mathfrak{H}_i = \tau_c - \tau_\epsilon^i + \tau_a^i & \text{else} \end{cases} \tag{16}$$

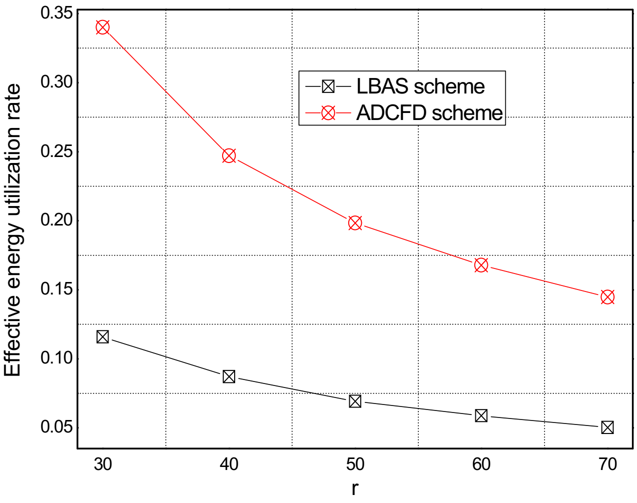

Fig. 21 the effective energy utilization rate under different r in two schemes

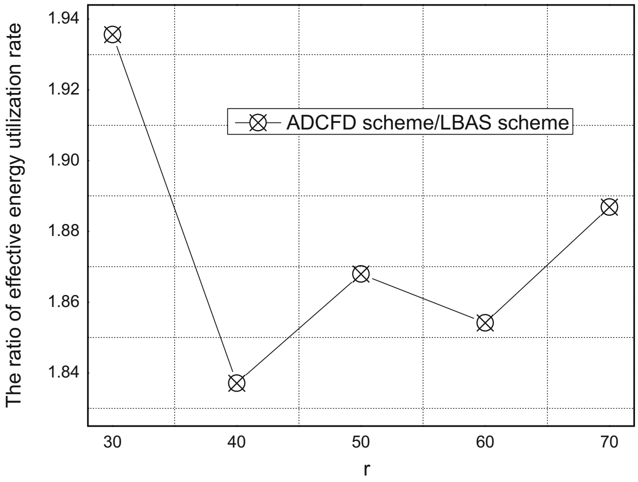

Fig. 22 the ratio of effective energy utilization rate under different r in two schemes

Proof. In the ADCFD scheme, the broadcast backbone is built based on the minimum covering node sets. The total minimum covering nodes is $|\Omega|$.

If one node wants to send program codes, the node senses the status of neighbor nodes. If some neighbor nodes are in the active time slots, program codes can be transmitted to neighbor nodes. If active time slots of neighbor nodes are early than active time slots of itself nodes, program codes can be transmitted to neighbor nodes in the next working cycle, or program codes are sent to neighbor nodes in this working cycle. For example, for node v_i in Ω, if $\tau_a^i < \tau_\epsilon^i$, the time for transmitting program codes is (τ_a^i -τ_{β_1}). But if $\tau_a^i > \tau_\epsilon^i$, the time for transmitting program codes is ($\tau_c - \tau_\epsilon^i + \tau_a^i$). Thus the transmission delay of the network is the time for transmitting program codes of nodes in the network. The total number of minimum covering nodes in the set is Ω. Thus, the number of transmission is as follows.

$$\gimel = \sum_{i \subseteq \Omega} \mathfrak{H}_i \qquad \begin{cases} \mathfrak{H}_i = \tau_a^i - \tau_{\beta_1} & \text{if} \quad \tau_a^i < \tau_\epsilon^i \\ \mathfrak{H}_i = \tau_c - \tau_\epsilon^i + \tau_a^i & \text{else} \end{cases}$$

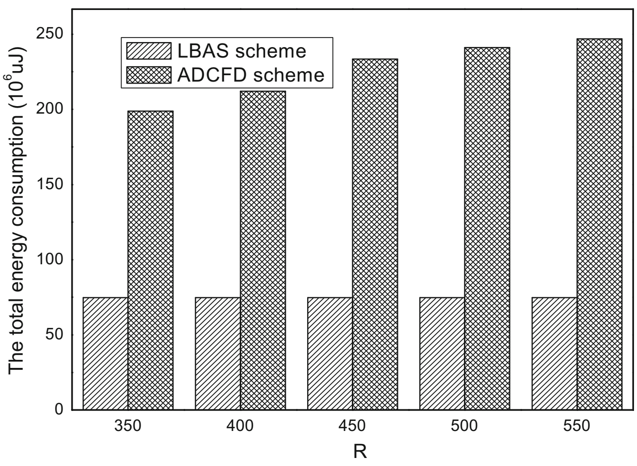

Fig. 23 the total energy consumption under different R in two schemes

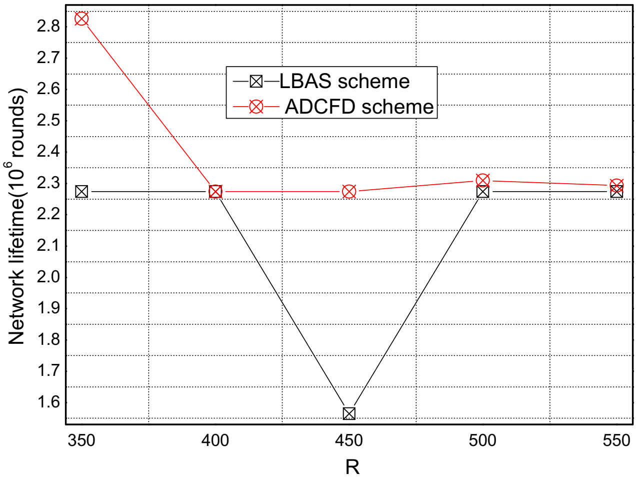

Fig. 24 the network lifetime under different *R* in two schemes

5.3 Analysis of energy and network lifetime

Considering node v_i at l m away from the sink in a data collection period τ_c, the energy consumption E^l of node v_i is as follows.

$$E^i = e^i_{sen} + \varpi^i_r Q^i_r + \varpi^i_s Q^i_s \tag{17}$$

Theorem 5. Considering node v_i at l m away from the sink in a data collection period τ_c, the initial energy of node v_i is E_{init}, the network lifetime l_i of node v_i is as follows.

$$l_i = E_{\text{init}} \Big/ \left(e^i_{sen} + \varpi^i_r Q^i_r + \varpi^i_s Q^i_s\right) \tag{18}$$

Proof. It can be seen that the energy consumption of node v_i at l m away from the sink is E^l, and the initial energy consumption of node v_i at l m away from the sink is E_{init}. The network lifetime is the rounds of transmitting data of node v_i. Thus, the network lifetime l_i of node v_i is as follows.

$$l_i = E_{\text{init}} \Big/ E^i = E_{\text{init}} \Big/ \left(e^i_{sen} + \varpi^i_r Q^i_r + \varpi^i_s Q^i_s\right) \qquad \square$$

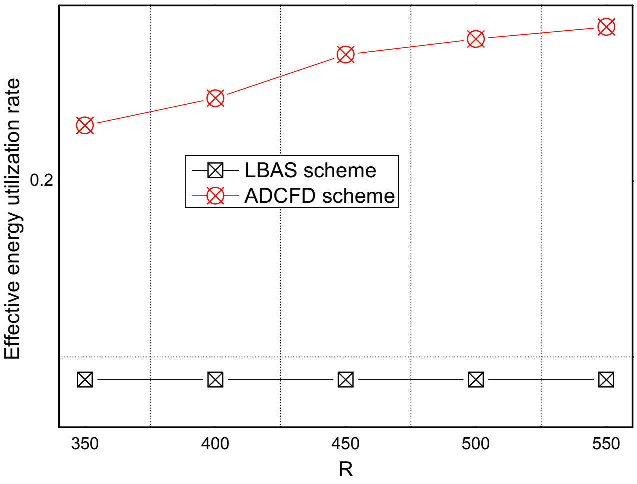

Fig. 25 the effective energy utilization rate under different *R* in two schemes

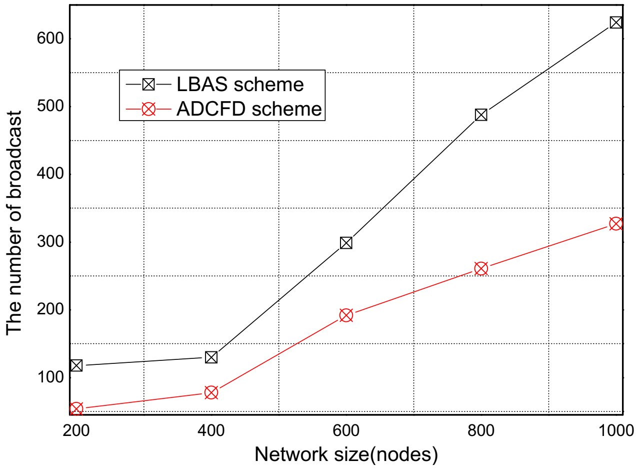

Fig. 26 The impact of network size with fixed value of |T| = 20 on the total number of broadcast

Inference 1. Considering effective energy utilization rate in the ADCFD scheme is φ_A, and effective energy utilization rate in the ADCFD scheme is φ_L. The ratio of effective energy utilization rate in the ADCFD scheme and effective energy utilization rate in the LABS scheme is as follows.

$$\psi = \frac{\varphi_A}{\varphi_L} = \frac{\sum\limits_{1\le i\le n}\left(e^i_{sea} + \varpi^i_{ra} Q^i_{ra} + \varpi^i_{sa} Q^i_{sa}\right)}{\sum\limits_{1\le i\le n}\left(e^i_{sel} + \varpi^i_{rl} Q^i_{rl} + \varpi^i_{sl} Q^i_{sl}\right)} \tag{19}$$

Proof: ϖ^i_{ra} is the power used by node v_i to receive a packet in the ADCFD scheme, e^i_{sea} is the energy consumption for sensing data in the ADCFD scheme, and ϖ^i_{sa} is the power

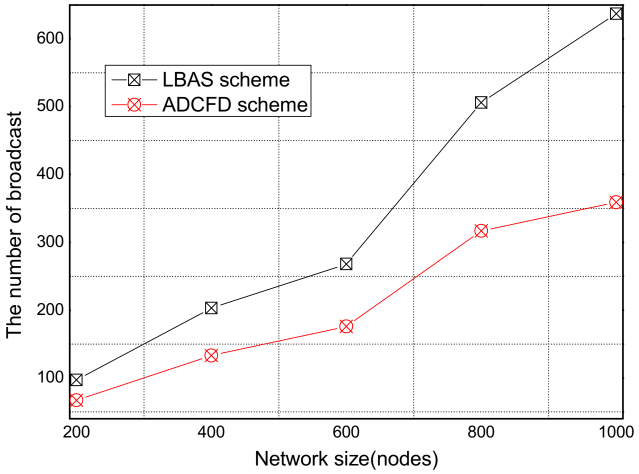

Fig. 27 The impact of network size with fixed value of |T| = 60 on the total number of broadcast

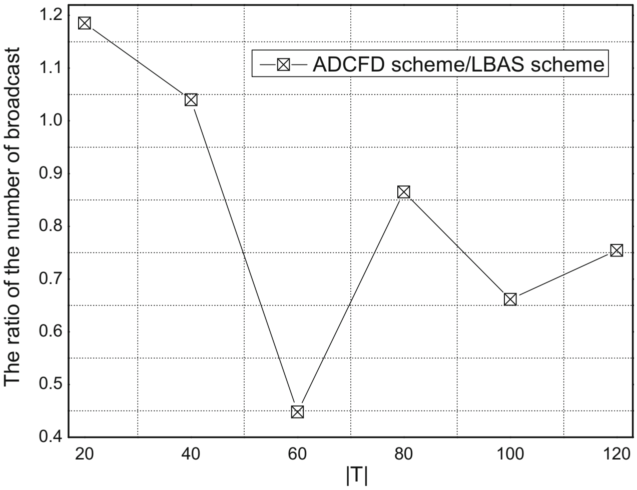

Fig. 28 The ratio of the number of broadcast for ADCFD scheme VS LBAS scheme with fixed value of |T| = 60

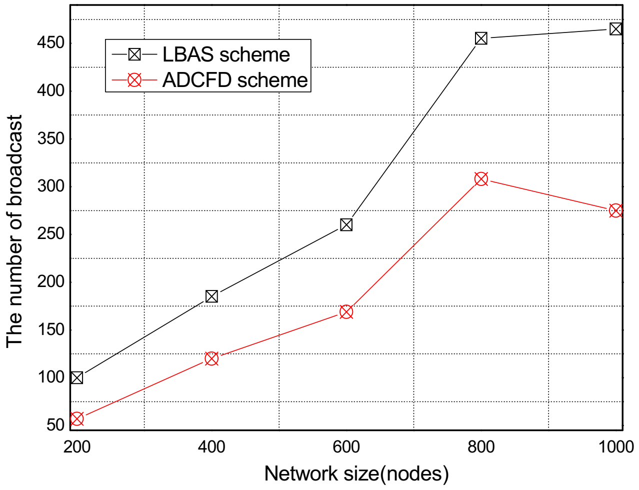

Fig. 30 The impact of network size with fixed value of |T| = 120 on the total number of broadcast

used by node v_i to send a packet in the ADCFD scheme. ϖ_{rl}^i is the power used by node v_i to receive a packet of node v_i in the LABS scheme, e_{sel}^i is the energy consumption in the sleep status in the LABS scheme, and ϖ_{sl}^i is the power used by node v_i to send a packet in the LABS scheme. It is calculated as follows.

$$\psi = \frac{\varphi_A}{\varphi_L} = \frac{\left(\sum_{1\le i\le n} e_A^i\right)/\left(\sum_{1\le i\le n} E_{\text{init}}\right)}{\left(\sum_{1\le i\le n} e_L^i\right)/\left(\sum_{1\le i\le n} E_{\text{init}}\right)}$$
$$= \frac{\sum_{1\le i\le n}\left(\varpi_{sea}^i + \varpi_{ra}^i Q_{ra}^i + \varpi_{sa}^i Q_{sa}^i\right)}{\sum_{1\le i\le n}\left(\varpi_{sel}^i + \varpi_{rl}^i Q_{rl}^i + \varpi_{sl}^i Q_{sl}^i\right)}$$

The energy consumption in two schemes is given in the Fig. 15. It can be seen that (1) the energy consumption of nodes near to the sink is higher than the energy consumption of nodes far from the sink. (2) The maximum energy consumption in the ADCFD scheme is the same as the maximum energy consumption in the LBAS scheme. (3) The energy consumption of nodes in the non-hotspots area in the ADCFD scheme is higher than the energy consumption of nodes in the non-hotspots area in the LBAS scheme. The reason is that the duty cycle of nodes in the ADCFD scheme is adjusted according to energy left of nodes. If nodes far from the sink have much energy left, the duty cycle of nodes can be increased a lot. In this paper, the duty cycle is the ratio of active period to the working period. The energy consumption may be higher if duty cycle of nodes is bigger. But the energy consumption of nodes near to the sink in the ADCFD scheme isn't higher than the energy consumption of nodes near to the sink in the LBAS scheme. It can be known that ADCFD scheme can retain network lifetime.

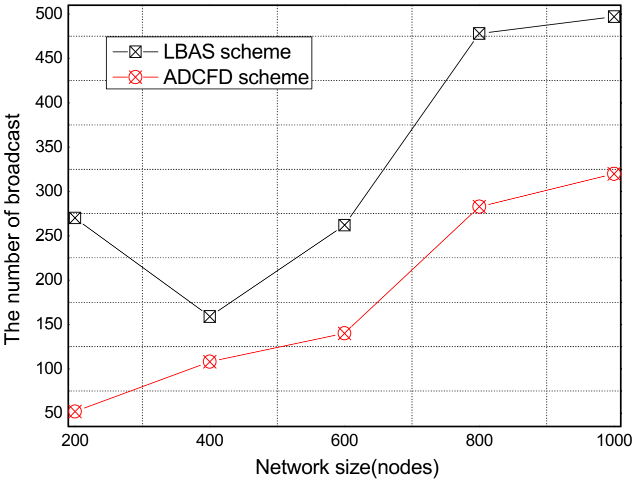

Fig. 29 The impact of network size with fixed value of |T| = 80 on the total number of broadcast

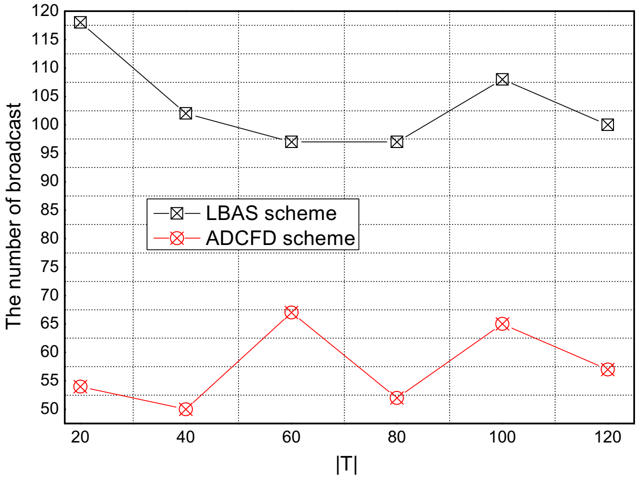

Fig. 31 The impact of |T| with fixed network nodes = 200 with fixed on the total number of broadcast

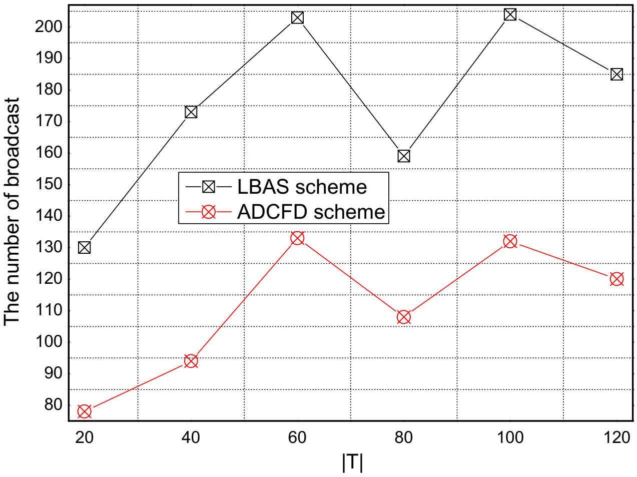

Fig. 32 The impact of |T| with fixed network nodes = 400 with fixed on the total number of broadcast

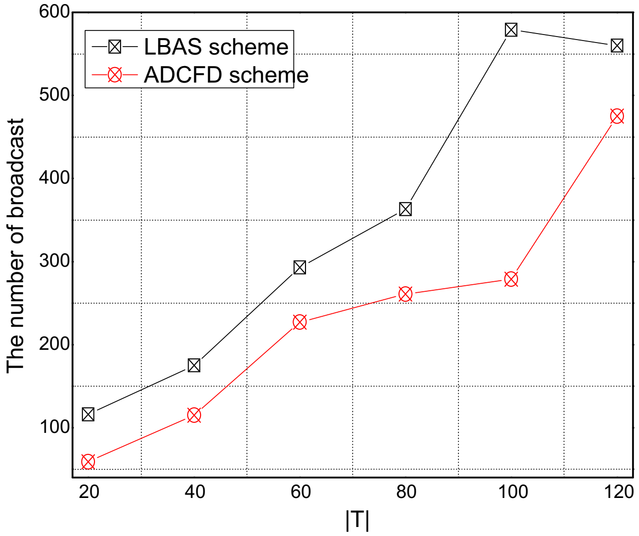

Fig. 34 The impact of |T| with fixed network nodes = 800 with fixed on the total number of broadcast

The energy consumption and network lifetime under different duty cycle in two schemes are given in the Fig. 16 and Fig. 17, respectively. The total energy consumption increases with the increases of duty cycle. Network lifetime in the ADCFD scheme isn’t smaller than network lifetime in the LBAS scheme, but the effective energy utilization in the ADCFD scheme is greater than that of the LBAS scheme (seen from Fig. 18). The reason is that in an ADCFD scheme, duty cycle of nodes far from the sink is increased according to energy left of nodes, and duty cycle of nodes near to the sink in the ADCFD scheme are the same as duty cycle of nodes near to the sink in the LBAS scheme. Thus, the effective energy utilization in the ADCFD scheme is greater than that of LBAS scheme. It shows that ADCFD scheme has better performance.

The energy consumption and network lifetime under different *r* in two schemes are given in Fig. 19 and Fig. 20, respectively. From Fig. 10 and Fig. 20, (1) the energy consumption decreases with the increase of *r*. (2) The total energy consumption in the ADCFD scheme is higher than the total energy consumption in the LBAS scheme. (3) Network lifetime in the ADCFD scheme isn’t lower than network lifetime in the LBAS scheme. The main invention of this paper is to increase duty cycle of nodes far from the sink. In the above analysis, there are much energy left of nodes far from the sink, the energy left of nodes is used to improve duty cycle of nodes. Thus, the energy consumption of nodes far from the sink in the ADCFD scheme is increased. But the energy consumption of nodes near to the sink in the ADCFD scheme isn’t higher than the energy consumption of nodes near to the sink in the LBAS scheme. Thus, the total energy consumption in the ADCFD scheme is higher

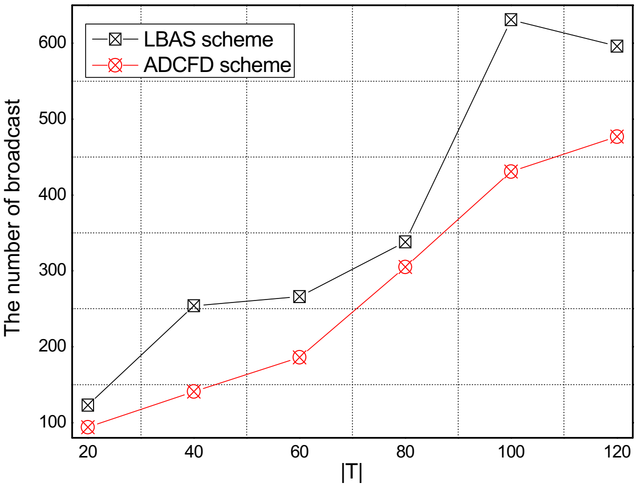

Fig. 33 The impact of |T| with fixed network nodes = 600 with fixed on the total number of broadcast

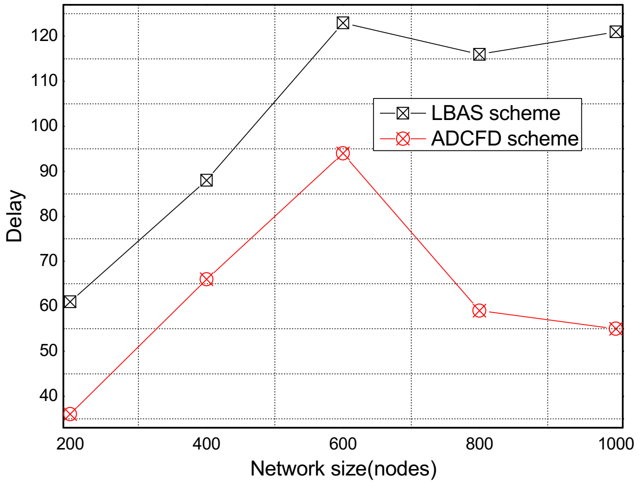

Fig. 35 The impact of network size with fixed value of |T| = 20 on the broadcast delay

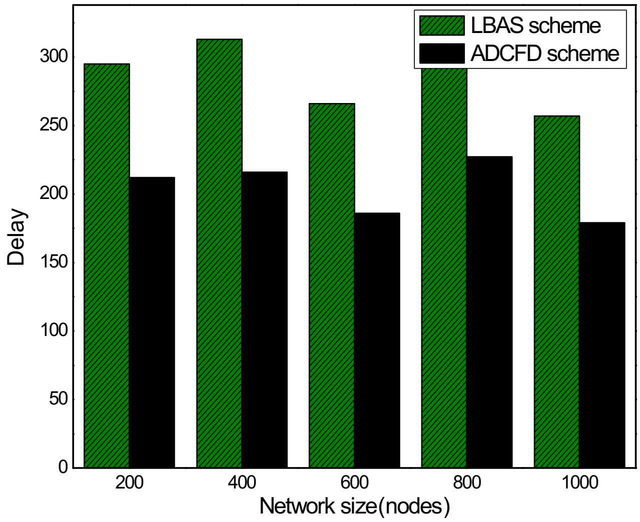

Fig. 36 The impact of network size with fixed value of |T| = 60 on the broadcast delay

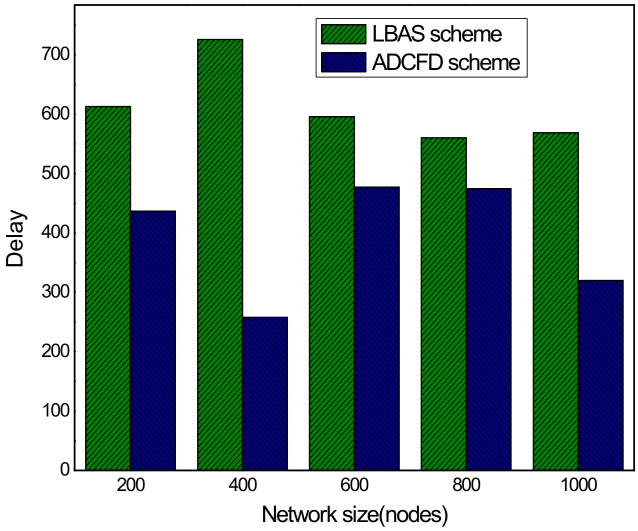

Fig. 38 The impact of network size with fixed value of |T| = 120 on the broadcast delay

than that of LBAS scheme, while the ADCFD scheme can retain the network lifetime.

The effective energy utilization rate is showed in Fig. 21. The effective energy utilization rate in the ADCFD scheme is higher than that of LBAS scheme. The improved effective energy utilization rate in the ADCFD scheme is 1.83713–1.93563 greater than that of LBAS scheme (see from Fig. 22).

The energy consumption and network lifetime under different *R* in two schemes are given in the Fig. 23 and Fig. 24, respectively. From Fig. 23 and Fig. 24, the total energy consumption in the ADCFD scheme is higher than the total energy consumption in the LBAS scheme. Network lifetime in the ADCFD scheme isn't smaller than network lifetime in the LBAS scheme. The effective energy utilization rate is showed in Fig. 25. The effective energy utilization rate in the ADCFD scheme is higher than that of LBAS scheme.

5.4 Experimental results and analyses

The experiments were conducted using Omnet++ platform, which is widely recognized in academia [22, 37]. OmneT++ is popular in academia for its extensibility (due to its open source model) and plentiful online documentation. Omnet++ represents a framework approach. Instead of containing explicit and hard-wired support for computer networks or other areas, it provides an infrastructure for writing such simulations. Without loss of generality, the network parameters are as follows: radius *R* =400 m, packet transfer radius *r* =50 m, event generation

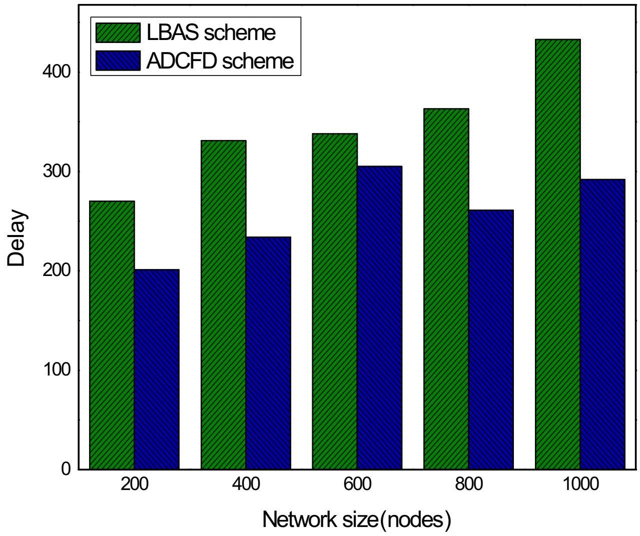

Fig. 37 The impact of network size with fixed value of |T| = 80 on the broadcast delay

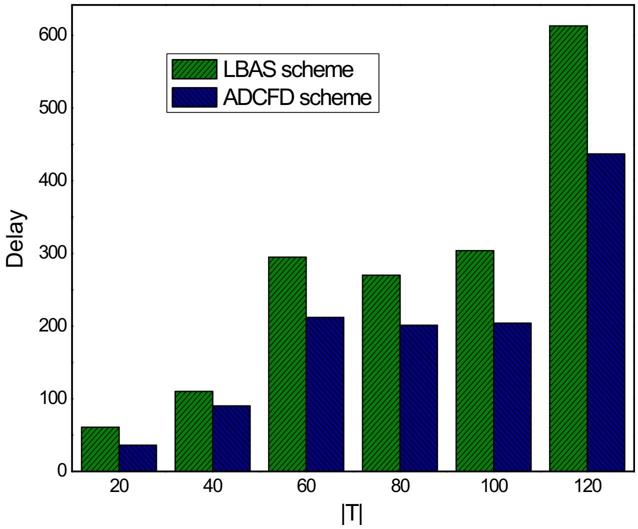

Fig. 39 The impact of |T| with fixed network nodes = 200 with fixed on the delay

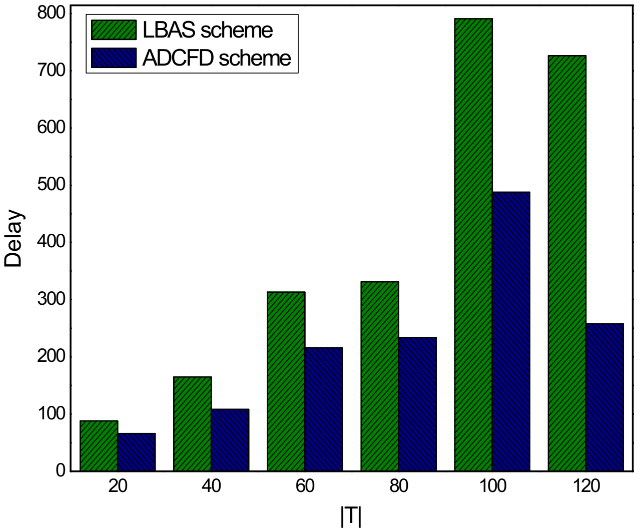

Fig. 40 The impact of |T| with fixed network nodes = 400 with fixed on the delay

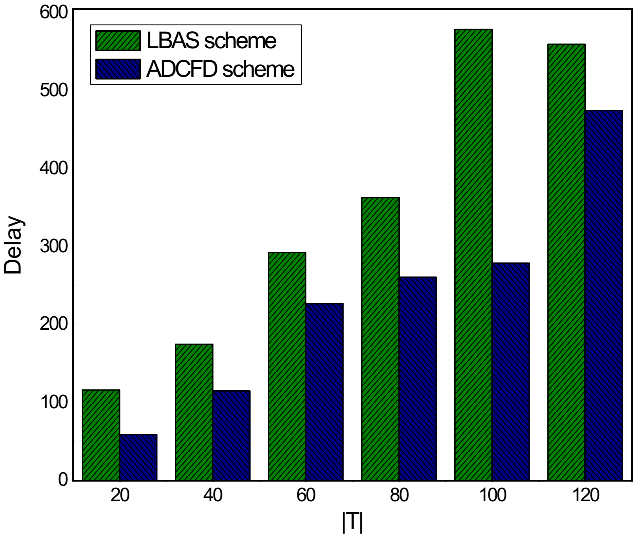

Fig. 42 The impact of |T| with fixed network nodes = 800 with fixed on the total number of transmissions

probability of node in one round λ =0.1, 1000 nodes are deployed, i.e., τ_a =0.1. Other parameters settings refer to Table 1.

5.5 Analysis of broadcast times

The number of broadcast under |T| = 20 and |T| = 60 are showed in Fig. 26 and Fig. 27, respectively. As can be seen, the number of transmission in ADCFD scheme is smaller than that of LBAS scheme. And the number of broadcast of two schemes generally increases with the network size increases. That is obvious because the |T| is determined in advance, with the number of node increases, a node may be selected as covering node at many duty cycle. Program codes is spread to all nodes in network, thus the number of broadcast increases. The number of transmission in the ADCFD scheme is improved by 44.776%- 118.519% (see from Fig. 28).

The total number of broadcast under |T| = 80 and |T| = 120 are given in Fig. 29 and Fig. 30, respectively. It can be seen that the number of broadcast in ADCFD scheme is smaller than that of LBAS scheme. The reason is that in the LBAS scheme, in order to make the number of transmission minimum, a broadcast tree is built. The first phase is to find the minimum covering node set, then constructing broadcast backbone. A broadcast backbone contains all covering nodes. A covering node has many neighbor nodes, those neighbor nodes have different active state. Thus, when program codes are spread to all nodes along the broadcast backbone, each node has different active status, one node will be broadcast several times to neighbor

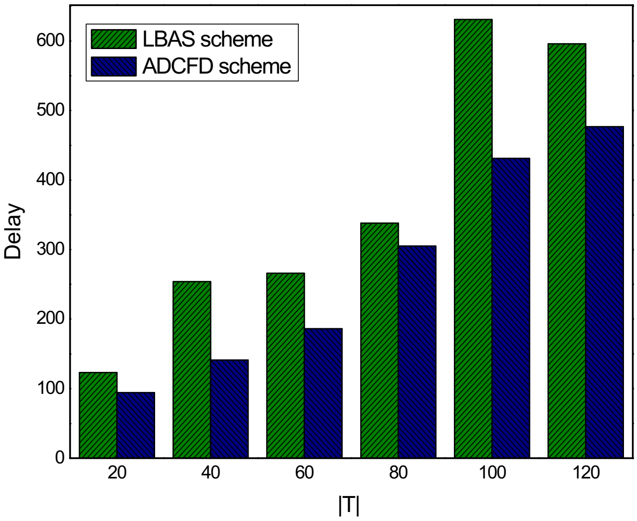

Fig. 41 The impact of |T| with fixed network nodes = 600 with fixed on the total number of transmissions

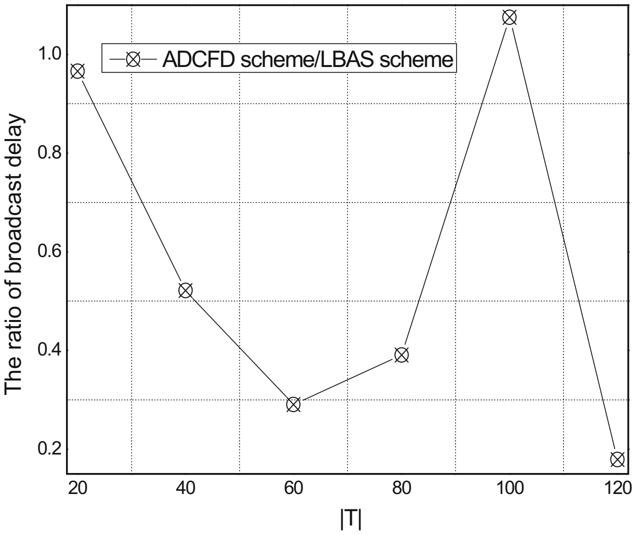

Fig. 43 The impact of |T| with fixed network nodes = 800 with fixed on the total number of transmissions

nodes. But duty cycle of nodes in the ADCFD scheme are adjusted, the covering sets are selected using the method as the LBAS scheme. The active time of neighbor nodes of a node is increased, and the active time is earlier. Thus one node can only broadcast once to broadcast program codes to all neighbor nodes to reduce the number of transmission.

The total number of broadcast under fixed 200 and 400 network nodes are showed in Fig. 31 and Fig. 32, respectively. Whether the network nodes is 200 or 400, the number of broadcast in ADCFD scheme is smaller than that of LBAS scheme. In Fig. 31, when |T| = 60, the number of broadcast in ADCFD scheme is 0.69072 times than that in LBAS scheme. But when |T| = 20, the number of broadcast in ADCFD scheme is 0.45763 times than that in LBAS scheme. In Fig. 32, when |T| = 40, the number of broadcast in ADCFD scheme is 0.54335 times than that in LBAS scheme. But when |T| = 80, the number of broadcast in ADCFD scheme is 0.67925 times than that in LBAS scheme.

The number of broadcast under fixed 600 and 800 network nodes are given in Fig. 33 and Fig. 34, respectively. Whether the network nodes is 600 or 800, the number of broadcast in ADCFD scheme is smaller than that of LBAS scheme. It shows that the ADCFD scheme can reduce the number of broadcast, it shows that ADCFD has better performance.

5.6 Analysis of broadcast delay

The broadcast delay under |T| = 20 and |T| = 60 are showed in Fig. 35 and Fig. 36, respectively. The broadcast delay in ADCFD scheme is smaller than that in LBAS scheme.

The impact of network size with fixed value of |T| = 80 and |T| = 120 on the broadcast delay are shown in Fig. 37 and Fig. 38, respectively. It can be seen that the delay in ADCFD scheme is smaller than that in the LBAS scheme.

The reason is that in the LBAS scheme, in order to make the number of broadcast minimum, a broadcast tree is built. The first phase is to find the minimum covering node sets, then build broadcast backbone. A broadcast backbone contains all covering nodes. A covering node has many neighbor nodes, those neighbor nodes have different active states. Thus, when program codes are spread to all nodes along broadcast backbone, one node will be broadcast several times to neighbor nodes due to different active states of neighbor nodes. But duty cycle of nodes in the ADCFD scheme are adjusted, the covering sets are selected using the method in the LBAS scheme. The active time of neighbor nodes of a node are adjusted, the active time of neighbor nodes of a node are the same, and the active time are earlier. Thus, one node can only broadcast program codes once to reduce broadcast time.

The transmission delay under fixed value of 200 and 400 network nodes are given in Fig. 39 and Fig. 40, respectively. As can be seen, the delay in ADCFD scheme is smaller than that of LBAS scheme. And the delay of two schemes generally increases as the network size increases. That is obvious because the |T| is determined in advance, with the number of node increases, a node may be selected as covering node at many duty cycle, and program codes is spread to all nodes in network, thus, the delay increases.

The transmission delay under fixed value of 200 and 400 network nodes and network nodes are shown in Fig. 41 and Fig. 42, respectively. It can be seen that the delay in ADCFD scheme is smaller than that of LBAS scheme. The reason is the same with the above. One node can only broadcast once to broadcast program codes to all neighbor nodes to reduce broadcast time. Transmission delay from dominating program codes is reduced by 17.895%- 107. 527% (see from Fig. 43).

6 Conclusion

There are many researches about Software defined wireless networks (SDWNs), it has great significance for configuring automatically and upgrading software in industrial and environmental monitoring fileds. The fast transmission for program codes becomes a important content. In the previous schemes, the methods are adopted to optimize scheduling algorithm in the newtork with the determined duty cycle, which reduce the transmission delay and the energy consumption. An adjustable duty cycle based fast disseminate (ADCFD) scheme is proposed for minimum-transmission broadcast (MTB) in smart wireless software-define sensor network. The difference of ADCFD scheme with the previous scheme is to increase the duty cycyle of nodes using energy left of nodes, thus, the transmission times and transmission delay are reduced while retaining network lifetime. The more important is that this method in this paper can be applied to many methods, which can improve the performance of previous schemes. Thus, it has a very good meaning.

Acknowledgments This work was supported in part by the National Natural Science Foundation of China (61379110, 61073104, 61379115, 61311140261, 61572528, 61272494, 61572528), the National Basic Research Program of China (973 Program) (2014CB046305) and Central South University undergraduate free exploration project (201510533283).

References

1. Lim H, Kim C (2001) Flooding in wireless ad hoc networks. Comput Commun 24(3–4):353–363
2. Liao X, Zheng L, He B et al (2015) A performance debugging framework for unnecessary lock contentions with record/replay

techniques. IEEE Trans Parallel Distrib Syst. doi:10.1109/TPDS.2015.2472412

3. Zeng D, Gu L, Guo S, Cheng Z et al (2016) Joint optimization of task scheduling and image placement in fog computing supported software-defined Embedded system. IEEE Trans Comput 65(12): 3702–3712
4. Wang K, Gavler A, Machuca C, Wosinska L et al (2016) Migration strategies for FTTx solutions based on active optical networks. IEEE Commun Mag 54(2):78–85
5. Sun L, Ren P, Du Q et al (2016) Fountain-coding aided strategy for Secure cooperative transmission in industrial. Wirel Sens Netw 12(1):291–300
6. Liu X, Dong M, Ota K, Hung P, Liu A (2016) Service pricing decision in cyber-physical systems: insights from game theory. IEEE Trans Serv Comput 9(2):186–198
7. He S, Shin D, Zhang J et al (2016) Full-view area coverage in camera sensor networks: dimension reduction and near-optimal solutions. IEEE Trans Veh Technol 65(9): 7448–7461
8. Xu Q, Su Z, Guo S (2016) A game theoretical incentive scheme for relay selection services in mobile social networks. IEEE Trans Veh Technol 65(8):6692–6702
9. Su Z, Xu Q, Fei M, Dong M (2016) Game theoretic resource allocation in media cloud with mobile social users. IEEE Trans Multimedia 18(8):1650–1660
10. Li H, Liu D, Dai Y, Luan T (2015) Engineering searchable encryption of mobile cloud networks: when qoe meets qop. IEEE Wirel Commun 22(4):74–80
11. Liu Y, Dong M, Ota K, Liu A (2016) ActiveTrust: Secure and trustable routing in wireless sensor networks. IEEE Trans Inf Forensics Secur 11:2013–2027
12. Li H, Yang Y, Luan T, Liang X, Zhou L, Shen X (2016) Enabling fine-grained multi-keyword search supporting classified subdictionaries over encrypted cloud data. IEEE Trans Dependable Secure Comput 13(3):312–325
13. He S, Chen J, Li X et al (2014) Mobility and intruder prior information improving the barrier coverage of sparse sensor networks. IEEE Trans Mob Comput 13(6):1268–1282
14. Wieselthier JE, Nguyen GD, Ephremides A. On the construction of energy-efficient broadcast and multicast trees in wireless networks. // INFOCOM 2000. Nineteenth Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. 2000:72–75
15. Das AK, Marks RJ, El-Sharkawi M et al (2003) R-Shrink: A heuristic for improving minimum power broadcast trees in wireless networks. In: Proc. of the IEEE GLOBECOM. Piscataway: IEEE Press, 2004:523–527.
16. Zhao D, Chin KW, Raad R (2014) Approximation algorithms for broadcasting in duty cycled wireless sensor networks. Wirel Netw 20(8):2219–2236
17. Liu A, Jin X, Cui G et al (2013) Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network. Inf Sci 230:197–226
18. Zhao S, Liu A (2017) High performance target tracking scheme with low prediction precision requirement in WSNs, International Journal of Ad Hoc and Ubiquitous Computing, http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijahuc
19. Liao X, Guo R, Yu D, Jin H, Lin L (2016) A phase behavior aware dynamic cache partitioning scheme for CMPs. Int J Parallel Prog 44(1):68–86
20. Long J, Dong M, Ota K, Liu A (2015) Green TDMA scheduling algorithm for prolonging lifetime in wireless sensor networks. IEEE Syst J. doi:10.1109/JSYST.2015.2448355
21. Le Duc T, Le DT, Zalyubovskiy VV et al (2016) Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks. Pervasive Mob Comput 27:116–132
22. Khiati M, Djenouri D (2013) BOD-LEACH: broadcasting over duty-cycled radio using LEACH clustering for delay/power efficient dissimilation in wireless sensor networks. Int J Commun Syst 28(2):296–308
23. Gu L, Zeng D, Guo S, Xiang Y, Hu J (2016) A general communication cost optimization framework for big data stream processing in geo-distributed data centers. IEEE Trans Comput 65(1):19–29
24. Li H, Lin X, Yang H, Liang X, Lu R, Shen X (2014) EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid. IEEE Trans Parallel Distrib Syst 25(8):2053–2064
25. Z. Chen, A. Liu, Z. Li, et al (2017) Energy-efficient Broadcasting Scheme for Smart Industrial Wireless Sensor Networks, Mobile Information Systems, doi:10.1155/2017/7538190.
26. Wang K, Du M, Yang D, Zhu C, Shen J et al (2016) Game-theory-based active defense for intrusion detection in cyber-physical Embedded systems. ACM Trans Embed Comput Syst 16(1):18
27. Chen Z, Liu A, Li Z, Choi YJ, Li J (2017) Distributed duty cycle control for delay improvement in wireless sensor networks. Peer-to-Peer Networking and Applications 10(3):559–578
28. Liu X, Liu A, Huang C (2017) Adaptive information dissemination control to provide Diffdelay for internet of things. Sensors 17(1): 138. doi:10.3390/s17010138
29. Deng X, Peng Q, He L, He T (2016) Interference-aware QoS routing for neighborhood area network in smart grid. IET Commun. doi:10.1049/iet-com.2016.0860
30. Duan X, Zhao C, He S, Cheng P, Zhang J (2016) Distributed algorithms to compute Walrasian equilibrium in mobile Crowdsensing. IEEE Trans Ind Electron. doi:10.1109/TIE.2016.2645138
31. Deng X, Li G, Dong M, Ota K (2017) Finding overlapping communities based on Markov chain and link clustering. Peer-to-Peer Networking and Applications 10(2): 411–420
32. Liao X, Jin H, Yu S et al (2015) A novel memory allocation scheme for memory energy reduction in virtualization environment. J Comput Syst Sci 81(1):3–15
33. Wang K, Shao Y, Shu L et al (2016) Mobile big data fault-tolerant processing for ehealth networks. IEEE Netw 30(1): 36–42
34. Liu X, Liu A, Deng Q, Liu H (2017) Large-scale programing code dissemination for software defined wireless networks. Comput J. doi:10.1093/comjnl/bxx014
35. Zeng D, Li P, Guo S, Miyazaki T et al (2015) Energy Minimization in multi-task software-defined sensor networks. IEEE Trans Computers 64(11):3128–3139
36. Deng X, He L, Li X, Liu Q, Cai L, Chen Z (2016) A reliable QoS aware routing scheme for neighbor area nework in smart grid. Peer-to-Peer Networking and Applications 9(4):616–627
37. OMNet++ Network Simulation Framework, http://www.omnetpp.org/

Xiao Liu Currently she is a student in School of Information Science and Engineering of Central South University, China. Her research interest is network computing.

Gaoxiang Li Currently he is a student in School of Information Science and Engineering of Central South University, China. His research interest is network computing.

Shigeng Zhang is an Associate Professor of School of Information Science and Engineering of Central South University, China. His major research interests are wireless sensor network and distributed computing.

Anfeng Liu is a Professor of School of Information 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.