An Energy-Efficient Routing Algorithm for Software-Defined Wireless Sensor Networks

Wei Xiang, *Senior Member, IEEE*, Ning Wang, and Yuan Zhou, *Member, IEEE*

***Abstract*— Recent significant research on wireless sensor networks (WSNs) has led to the widespread adoption of software-defined WSNs (SDWSNs), which can be reconfigured even after deployment. In this paper, we propose an energy-efficient routing algorithm for SDWSNs. In this algorithm, to make the network to be functional, control nodes are selected to assign different tasks dynamically. The selection of control nodes is formulated as an NP-hard problem, taking into consideration of the residual energy of the nodes and the transmission distance. To tackle the NP-hard problem, an efficient particle swarm optimization algorithm is proposed. Simulation results show that the proposed algorithm performs well over other comparative algorithms under various scenarios.**

***Index Terms*— SDWSNs, sensing tasks, control nodes, residual energy, transmission distance, PSO.**

I. Introduction

THE emergence of big data and cloud technology has drive a fast development of wireless sensor networks (WSNs) [1]–[4]. A sensor node is normally comprised of one or more sensor units, a power supply unit, a data processing unit, data storage, and a data transmission unit [5]. A wireless sensor network is a collection of wireless nodes with limited energy that may be mobile or stationary and are located randomly in a dynamically changing environment. Wireless sensor networks hold the promise of revolutionizing the way we observe and interact with the physical world in a wide range of application domains such as environmental sensing, habitat monitoring and tracking, military defence, etc. [6]. The characteristics of low-cost, low-power, and multifunctional sensor have attracted a great deal of research attention, in that sensor nodes can perform intelligent cooperative tasks under stringent constrains in terms of energy and computational resources.

However, most previous research work only considers the scenario where a WSN is dedicated to a single sensing task, and such application-specific WSN is prone to high deployment costs, low service reutilization and difficult hardware recycling [7]. A software-defined wireless sensor network (SDWSN) consists of software-defined sensor nodes that can dynamically reconfigure their functionalities and properties by loading different programs on-demand according to real-time sensing requests. SDWSNs are emerging as a compelling solution to tackle the above issues. A software-defined sensor node equipped with several different types of sensors is able to undertake a variety of sensing tasks according to deployed and activated programs. In recent years, especially due to the advent of forthcoming 5G networks [8]–[11], a number of prototypes have been practically implemented. SDWSNs enable programmable control in network and virtualization of network equipment by decoupling the control plane and data plane [12]. In SDWSNs, control intelligence is taken out from data plane devices and implemented in a logically centralized controller (network operating system, however can be formed by distributed clusters), which interacts with data plane devices through standard interfaces. Network operators run software programs on the controller to automatically manage data plane devices and optimize network resource usage [13]. This architecture enables up-to-date control schemes to be developed and deployed so as to enable new smart sensing services, making simplified network management in WSNs, which makes the future of SDWSNs bright [14].

However, to realize the aforementioned advantages of SDWSNs is not without challenges [15]. In a sensor network, each node acts as both a sensor and router, with limited computing and communications capabilities, and storage capacity. However, in many WSN applications, the deployment of sensor nodes is performed in harsh environments, which makes sensor replacement difficult and expensive [16]–[18]. Thus, in many scenarios, wireless nodes must operate without battery replacement for a long period of time. Consequently, the energy constraint is vital for the design of WSNs [19] and SDWSNs. In an SDWSN, although different virtual networks can work together on top of the same physical infrastructure, the centralized control plane may lead to high energy costs due to information collection to reach a global view, and multiple virtual networks may compete for common physical network resources. Therefore, resource utilization of the SDWSN also needs to be carefully designed.

In this paper, we consider the SDWSN as illustrated in Fig. 1, which consists of a sensor control server and a set of software-defined sensor nodes. The large scale of deployed nodes that are equipped with multi-functions are able to execute multi-tasks simultaneously. For example,

Manuscript received May 1, 2016; revised June 8, 2016; accepted June 9, 2016. Date of publication July 22, 2016; date of current version September 16, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61201179 and Grant 61571326. The work of W. Xiang supported by the Queensland Government of Australia through the Smart Futures Fellowship. The associate editor coordinating the review of this paper and approving it for publication was Dr. Yin Zhang.

W. Xiang is with the College of Science and Engineering, James Cook University, Cairns, QLD 4870, Australia.

N. Wang and Y. Zhou are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: zhouyuan@tju.edu.cn).

Digital Object Identifier 10.1109/JSEN.2016.2585019

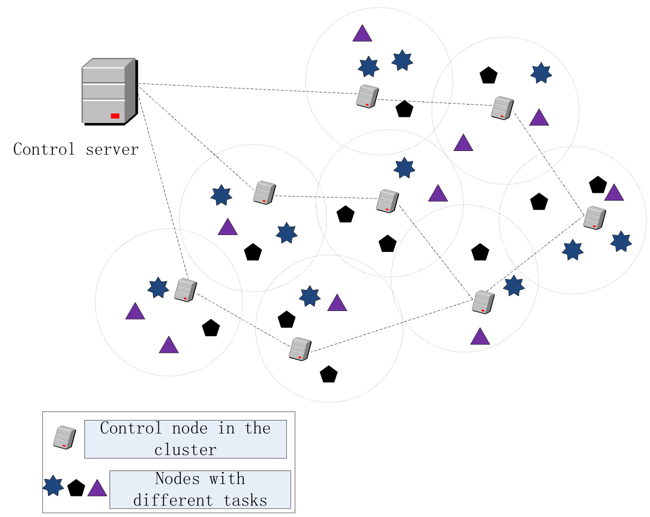

Fig. 1. An example of the software-defined sensor network with multi-tasks.

a software-defined node can monitor the temperature and humidity at the same time. The sensor control server can reprogram some sensor nodes by distributing a corresponding program to them for the tasks. We divide the sensor nodes into clusters, each of which consists of a control node and a number of common nodes with different tasks, aiming at balancing the energy consumption of sensor nodes and avoiding the collision of data transmission. Like TinyOS [20], every SDWSN OS can support multi-tasking that executes independently and non-preemptively.

Traditional routing protocols in WSNs consume more energy for multi-tasking sensor networks because of the inflexibility. Therefore, based on the above architecture, we propose a new energy-efficient routing algorithm for software-defined wireless sensor networks. The control server selects the control nodes of each cluster, and the control nodes instruct the intra-cluster nodes to complete different tasks. In this paper, we are motivated to investigate how to minimize the energy consumption if reprogramming by considering the control nodes' selection and multicasting routing. Our main contributions are summarized as follows:

- We propose an energy-efficient routing algorithm for the multi-tasking SDWSNs. The selection of control nodes is formulated as an NP-hard problem, taking into account the residual energy of the nodes and the transmission distance; and
- To tackle the NP-hard problem, we propose an efficient particle swarm optimization (PSO) algorithm to solve it.

The remainder of this paper is organized as follows. Section II briefly reviews related studies. The network model under consideration is described in Section III. A new routing algorithm based on non-linear weight particle swarm optimazation (NWPSO) is proposed in Section IV. Section V shows our performance evaluation. Finally, Section VI draws the concluding remarks.

II. Related Work

A. Overview of Software-Defined Wireless Sensor Network

The software-defined wireless sensor network represents a new paradigm shift that offers a significant promise to

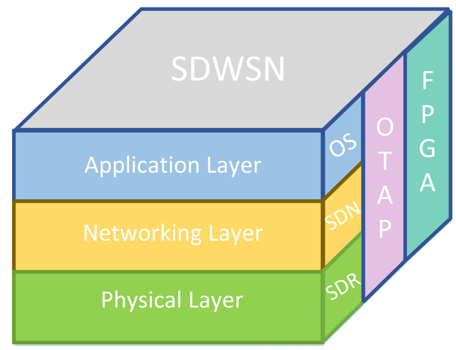

Fig. 2. A logical view of the SDWSN.

ubiquitous sensing and sensory data access through sensing-as-a-service. Fig. 2 illustrates the logical view of the SDWSN. There are a few pioneering investigations that have been studied in the literature.

Lecointre *et al.* [21] propose a software-defined radio interface for wireless sensor networks. Rossi *et al.* [22] present a system dubbed SYNAPSE++ for over-the-air reprogramming of wireless sensor networks. In [40], a die-hard sensor network is described, which can automatically monitor a disaster-hit region by scattering many sensor nodes in the region. A TinyOS-based SDN framework that allows for multiple controllers within the WSN is presented in [23], which is hardware independent. Huang *et al.* [24] propose a SDWSN prototype to improve the adaptability of WSNs for environmental monitoring applications, taking account of some constraints.

The above existing works have demonstrated the feasibility of the SDWSN. Different from the above studies, this paper focuses upon the energy efficiency in the network layer of the SDWSN, as shown in Fig. 2, as well as a particular emphasis on multi-task scheduling.

B. WSN Routing Algorithm

Routing is important in the WSN in determining the optimum routing paths of data packets, and there have been a great number of popular routing algorithms for the WSN.

Ad hoc On-demand Distant Vector (AODV) [25] was proposed in 1999, and became an IETE standard. It is a routing algorithm in consideration of the distance between the nodes. Its quick adaption to link conditions, low memory usage and low network utilization make the ADOV algorithm popular. However, the number of flooding messages increases significantly thanks to the increasing routing request messages. Clustering protocols can aid in data aggregation through efficient network organization. Low-energy adaptive clustering hierarchy (LEACH) [26] is one of the most well-known WSN hierarchical routing algorithms, which selects the cluster headers (CHs) based on a predetermined probability in order to rotate the CH role among the sensor nodes and to avoid fast depletion of the CH's energy. LEACH operates in two phases, i.e., the cluster setup phase and the steady phase. In the cluster setup phase, the cluster heads are selected and then broadcast to other nodes. In the steady state phase, actual transmission of data occurs. However, the study of LEACH considers only energy consumption in receiving the advertisements from the CHs at each sensor node during the setup phase. The number of the cluster heads varies and the CHs do not have a good distribution. Furthermore, LEACH requires the transmission between the cluster heads and the sink to be completed in a single hop, which consumes a large quantity of energy and disrupts the energy balancing of nodes if the CHs are located far away from the sink. In [27], DF-LEACH is proposed as an improvement of LEACH, which takes into account the distance of the CH to the sink node, and thus saves communications energy. In [28], a hybrid energy-efficient distributed clustering approach (HEED) is proposed. The initial probability for each sensor to become a cluster head is dependent of its residual energy, and the performance results are fairly good. Hausdorff [29] uses a greedy algorithm to select the cluster heads based on residual energy and location information, and this method can significantly prolong the network lifetime. In [30], an unequal cluster-based routing protocol is proposed, which focuses on load balancing in order to address the hot-spot issue. Mottola *et al*. [31] propose an adaptive energy-aware multi-sink routing algorithm, which is expressly designed for many-to-many communications. In [32], the authors address the issue of load balancing through considering different hop distances for the clusters. EDIT [33] is proposed to select the cluster head based on not only energy but also delay.

The traditional routing algorithms are unable to adapt to the flexibility of SDWSNs. Consequently, we propose a new routing algorithm for the SDWSN, which can accommodate the SDWSN's conditions flexibly and helps achieve better results.

III. System Model

A. Network Model

In this paper, we consider the network architecture as shown in Fig. 1. $G = (V, L)$ denotes the directed graph representing the network. V is the vertex set, including one control sever and a number of sensor nodes distributed within the monitoring field randomly. L is the set of directed links. The following assumptions on the sensor network and sensor nodes under consideration in this paper are made:

- We consider a set of λ sensing targets, e.g., temperature, humidity, and so on, which are randomly distributed within the same region of the SDWSN;
- The resources in a sensor node should be managed, controlled and allocated in an orderly manner in support of various sensing tasks. Besides, to complete different tasks, corresponding programs are stored on the sensor nodes, and the sensor node shall allow application programmers to adjust the sensor functionalities via invoking different programs;
- Each sensor node has the same ability to operate either in the sensing mode to perceive the environmental parameters or in the communications mode to send data among each other, or directly to the control server, and each node can gather data packets from a cluster member when acting as the control node. And each sensor node is assigned a unique identifier (ID);
- The sensor nodes and control server are stationary after deployment, which is typical for sensor network applications;
- Initial energy is fair to each sensor node, and the network is considered homogeneous;
- All the nodes are left unattended without battery replacement after deployment;
- Nodes are location-unaware, i.e., not equipped with GPS-capable antennae or other similar equipment, and each node is assigned a number according to its location;
- The links between the nodes are symmetric. A node can estimate the distance to another node based only on the received signal power;
- The control server is externally powered.

B. Energy Consumption Model

A simplified model is considered in this paper for communications energy consumption in consideration of path losses. Both the free space (d^2 power loss) and multipath fading (d^4 power loss) channel models are employed [34], depending on the distance between the transmitter and receiver. Power control can be used to compensate for this loss. If the distance is less than a threshold d_0, the free space model is used; otherwise, the multipath model is adopted. The required energy for transmitting a one-bit packet over distance d is

$$E_{TX}(l,d) = \begin{cases} k \times E_{elec} + k \times E_{fs} \times d^2, & d \le d_0 \\ k \times E_{elec} + k \times E_{mp} \times d^4, & d > d_0. \end{cases} \quad (1)$$

And for the control nodes, the energy for fusing a k-bit packet is given by

$$E_{TX}(l,d) = \begin{cases} k \times (E_{elec} + E_{DA}) + k \times E_{fs} \times d^2, & d \le d_0 \\ k \times (E_{elec} + E_{DA}) + k \times E_{mp} \times d^4, & d > d_0 \end{cases} \quad (2)$$

where E_{TX} is the transmission energy, E_{DA} is the energy consumed for data aggregation, d is the distance between two nodes or between a node and the sink, E_{elec} is the energy dissipated per bit to run the transmitter or receiver circuit, which depends on factors such as digital coding, modulation, filtering, and signal spreading. E_{fs} and E_{mp} depend on the transmitter amplifier model, k is the length of the data transmitted, d_0 is the transmission distance threshold, and usually computed as

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}. \quad (3)$$

To receive a k-bit message, the radio consumes the following amount of energy

$$E_{RX}(k) = k \times E_{elec}. \quad (4)$$

IV. Routing Algorithm Based on NWPSO

A. NWPSO Algorithm

Particle swarm optimization (PSO) [35] is a population-based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by the social behavior

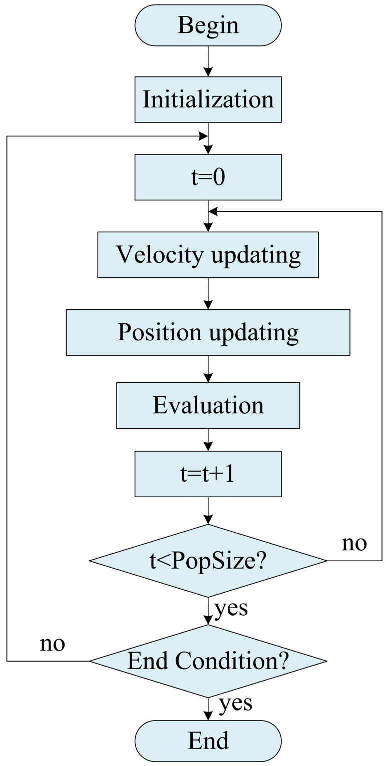

Fig. 3. Flowchart of the PSO algorithms.

of birds flocking or fish schooling. PSO begins with a group of random particles (random solutions), aiming at finding out the optimum solution through an iterative process. Each particle has a fitness value, which will be evaluated by the fitness function to be optimized in each generation. During the search process, each particle in the population consists of a d-dimensional vector including the velocity vector $v_i = [v_{i1}, v_{i2}, ..., v_{id}]$, the current position vector (pBest) $x_i = [x_{i1}, x_{i2}, ..., x_{id}]$, and the previous best position vector $p_i = [p_{i1}, p_{i2}, ..., p_{id}]$, where d is the dimensionality of the search space. What's more, the whole population maintains a global best-so-far population vector $p_g = [P_{g1}, p_{g2}, ..., p_{gd}]$ [36]. The flowchart of PSO is shown in Fig. 3. As can be seen from the figure, during each iteration of the evolutionary process in PSO, each particle learns from its own search experience pBest and the swarm's search experience gBest to update its velocity v_i and position x_i [35], [36]. During the iterations, the velocity of the particle is updated according to the following

$$v_{id}(t+1) = wv_{id}(t) + c_1\alpha(p_{id} - x_{id}(t)) + c_2\beta(p_{gd} - x_{id}(t)). \tag{5}$$

The position of the particle is updated as follows

$$x_{id}(t+1) = x_{id}(t) + v_{ij}(t) \tag{6}$$

where the representation of v_{id} is similar to that of x_{id}, P_{id} and P_{gd} are the dth dimension of the ith particle's velocity. Coefficients α and β are two randomly generated values within the range of $[0, 1]$ for the dth dimension. c_1 and c_2 are two acceleration parameters which are commonly set to 2.0 or adaptively controlled according to the evolutionary states. Factor w is the inertial weight, which plays the role of controlling the impact of the previous velocity of a particle on the current one so as to balance between global search (large inertial weight) and local search (small inertial weight).

However, PSO exhibits poor local search ability and often leads to premature convergence, especially in complex multi-peak search problems [37]. To tackle this issue, this paper proposes a method which adapts itself nonlinearly as follows

$$w = (w_{max} - w_{min} - d_1) \times e^{\frac{1}{1+d_2 \times t/K}} \tag{7}$$

where w_{max} and w_{min} represent the maximum and minimum inertial weights and are always set to 0.9 and 0.4, respectively. K is the maximum number of allowed iterations while t represents the current iteration. d_1 and d_2 are two control factors which control the value of w between w_{min} and w_{max}.

The execution of the algorithm is comprised of two phases, i.e., the control nodes' selection phase and the data transmission phase. The two phases are performed in each round of the network operation and repeated periodically. We elaborate on how to use the non-linear weight particle swarm optimization algorithm (NWPSO) to select the control nodes in the next section.

B. Control Nodes' Selection

The selection of control nodes is the most important procedure in our algorithm, which is implemented by the control server. The control nodes assign the tasks to the intra-cluster nodes, and collect the data from the cluster before transmitting them to the control server. Therefore, the control nodes consume much more energy compared to other nodes. As a result, the control server tends to select the control nodes with higher residual energy and better location according to the location of the control server among the sensor nodes, and then forms clusters with an equal distribution of the sensor nodes based on their locations and residual energy. This problem can be seen as an NP-hard problem. As such, we propose an NWPSO algorithm to identify the optimal control nodes.

We define the energy information as the fitness function f_1, which is the ratio of the control nodes' average residual energy to the common nodes' average residual energy in the current round. As a result, maximizing f_1 means that the nodes with higher energy should be selected as control nodes.

The average node residual energy can be expressed as

$$E_{node} = \sum_{i=1}^{N} E_i^{res}/|N| \tag{8}$$

where E_i^{res} is the node residual energy, and N is the number of nodes. We define E_i^{res} as the average residual energy of common nodes, and E_j^{res} as the average residual energy of control nodes.

Meanwhile, the average distance between the sensor nodes and the control server can be expressed as

$$d_{node} = \sum_{i=1}^{N} \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2}/|N| \tag{9}$$

Algorithm 1: Control Nodes' Selection Algorithm

Initialization:
repeat
  for each particle **do**
    Calculate the distance parameters and energy parameters;
    Compute the value of the objective function as per (10);
    Change the weight as per (7);
    Update the velocity and position as per (5) and (6);
    Update the control nodes.
  end for
until the maximum number of iterations is reached
Define the selected node as 'C'

where (x_s, y_s) is the coordinate of the control server, while (x_i, y_i) is the coordinate of the sensor nodes. Denote by d_{common} the average distance between the common nodes and the control server, and $d_{control}$ the average distance between the control nodes and the control server. Mreover, another fitness function f_2 can be expressed as the ratio of the average distance between the common nodes and the control server to that between the control nodes and the control server.

To select the nodes with higher residual energy and better location, we define the following fitness function

$$f = \gamma f_1 + \delta f_2 = \gamma \frac{E_{control}}{E_{common}} + \delta \frac{d_{common}}{d_{control}} \tag{10}$$

where

$$\gamma + \delta = 1. \tag{11}$$

By maximizing the object function f, it is expected that the cluster formation and the control nodes' selection of the WSN can be optimized so as to improve the energy efficiency of the sensor network. If a node has more residual energy and is closer to the control server, it is more likely to be selected as a control node. We set $\gamma = \delta = 0.5$ to trade off the same contribution of residual energy and location in selecting the control nodes. Algorithm 1 describes the selection of control nodes in detail.

In a practical environment, sensor nodes are battery powered. The residual energy of node E_i^{res} can be represented by its present battery voltage. The transmission of data packets from the common nodes to its control nodes will reduce the residual energy of the nodes. The location of the nodes can be obtained by implementing localization services as detailed in [38].

C. Clustering Information

After the control nodes are selected using the probabilities in Sections IV.B, each control node introduces itself to the network by broadcasting a small advertisement message (ADV), which uses a non-persistent carrier-sense multiple access (CSMA) MAC protocol. The message includes the control node's ID and a header that distinguishes this message as an advertisement message. Each common node determines its cluster by choosing the control node that requires the minimum transmission communications energy, based on the strength of the ADV message from each cluster head. Afterwards, a random cluster is chosen. After each common node has decided as to which cluster it belongs to, it must inform the control nodes of its decision by transmitting a JOIN-REQ message. The message is again very short, consisting of the node's ID, the belonging control node's ID and the sender's residual energy. In this way, the clusters are formed, and the duty of each node in the network is determined.

The control node in a cluster acts as the control center to coordinate data transmission. The control node assign different tasks to the common nodes in its cluster randomly. And each task has a start order coded by the control node. Then the control node sets up a TDMA scheduler and broadcasts the SCHEDULE-MSG message, which includes the TDMA scheduler and the start order with the common node's ID to start its assigned task. This helps avoid collisions among data messages, and also allows the radio components of each common node to be turned off at all times, except during the common nodes' transmission time, which uses limited bandwidth. This enables us to increase spectral efficiency and to decrease energy consumption consumed by individual sensors. Once the TDMA scheduler and the assigned tasks are known by all the common nodes, the common nodes start their tasks. Then the clustering setup phase is complete, and the data transmission phase begins at the same time with a determined topology.

In the data transmission phase, the common nodes send data to their control node according to the TDMA scheduler. The nodes are synchronized at all times through the control server sending out synchronization pulses to the nodes. The control nodes must be awake all all times to receive all the data from the common nodes within the cluster, and then aggregate the data to enhance the common signal and reduce the uncorrelated noise among the signals. Afterwards, the control node transmits the aggregated data to the control server.

V. Simulations and Results

In this section, the performance of our proposed method is evaluated via computer simulations. The network lifetime is the time span from deployment to the instant, when the network is considered nonfunctional [39]. In periodic data collection applications, the proper definition of lifetime is the time span between the start of network operation and the time when first node the dies. The simulation models and programs are developed in MATLAB.

The NWPSO algorithm has been applied to find the optimal control nodes for the SDWSN. The convergence of the fitness value for the objective function given in (12) over the number of generations using NWPSO is shown in Fig. 4. As can be observed from Fig. 4, the fitness value converges after less than 30 iterations.

Firstly, we compare the proposed routing algorithm with LEACH in terms of the number of survival nodes over time. As time goes by, the energy of each sensor node is consumed in the process of completing the assigned tasks, and a node is considered dead when its energy is depleted. We change the number of control nodes as shown in Fig. 5, the x-axis of

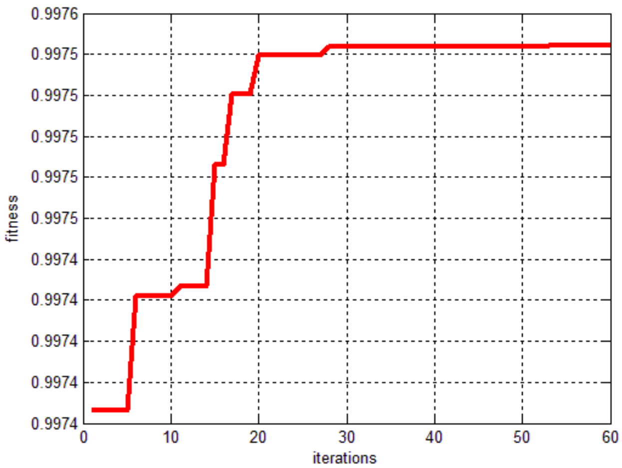

Fig. 4. Convergence of the objective function when using NWPSO.

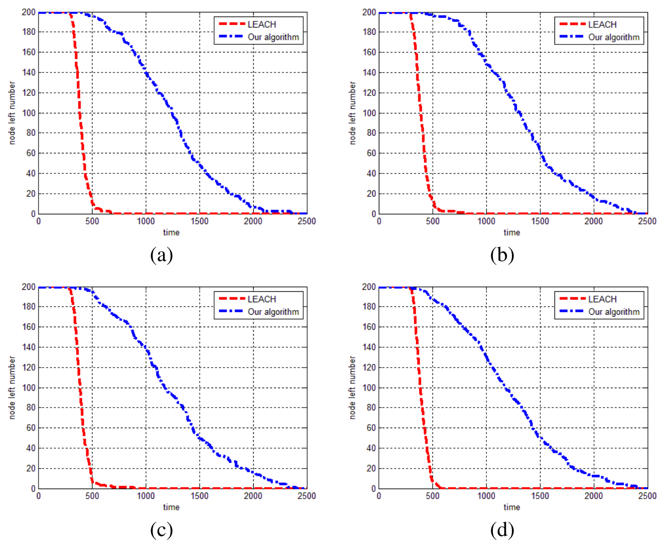

Fig. 5. Performance variance with different numbers of control nodes. From left to right and top to bottom: (a) 12 control nodes; (b) 14 control nodes; (c) 16 control nodes; and (d) 18 control nodes.

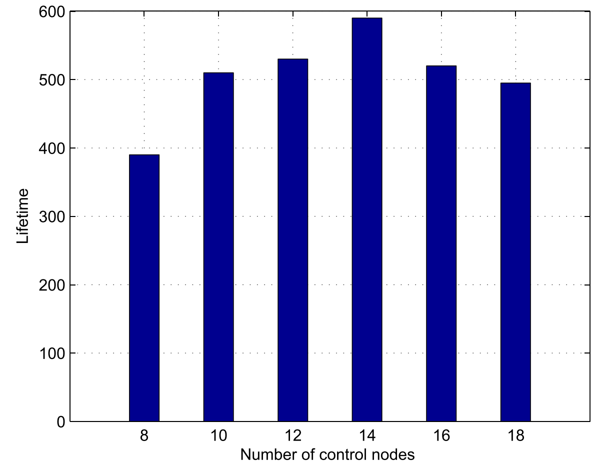

Fig. 6. Lifetime with different numbers of control nodes.

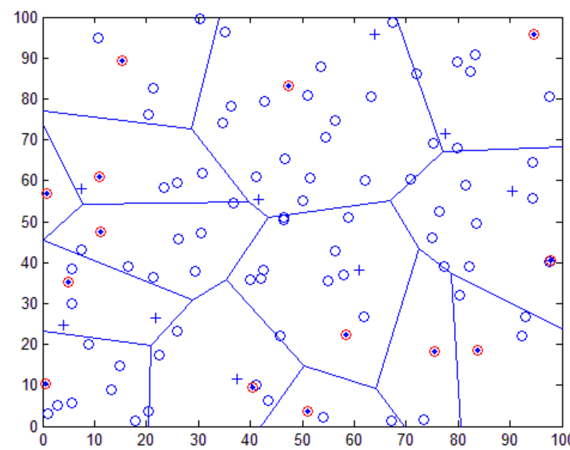

Fig. 7. Cluster formation for 14 control nodes.

figures is the simulation time while the y-axis is the number of survival nodes. Our algorithm consistently outperforms LEACH. This is because that the proposed algorithm takes the nodes' residual energy and location into consideration when selecting the control nodes, and the control nodes assign the tasks dynamically. This helps achieve better performance over LEACH.

Fig. 6 plots the lifetime variance with different control nodes. The lifetime is longest when the number of control nodes is 14. As can be seen from Figs. 5 and 6, the optimum number of control nodes is 14 in this situation. Increasing the number of control nodes will lead to reduced energy consumption of the intra-cluster, which can achieve much more evenly distributed control nodes. However, this does not mean that we can increase the number of control nodes to prolong the network lifetime, which is due to two reasons. Firstly, the control server is far from the network, if more nodes are selected as the control nodes, more control nodes will communicate with the control server, and thus will increase the overall transmission distance between the control nodes and the control server. As a result, more energy will be consumed by the control nodes. Secondly, one of the duties of the control nodes is to aggregate data to reduce the redundant information of common nodes. If we use more control nodes, two similar data may be transferred to different control nodes. More redundant information is transmitted to the control server consequently. So more energy is consumed in transmitting information. In conclusion, there exits an optimum number of control nodes to maximize the network efficiency. Based upon our simulations, 14 control nodes can achieve better results. Fig. 7 plots the topology of the network with 14 control nodes.

HEED is a data-gathering algorithm for the WSN, which is based on the candidate's residual energy and the proximity to its neighbors. Hausdorff clustering is also an efficient algorithm based on the nodes' residual energy and the position of the nodes. To assess the performance of our algorithm with different node densities, HEED, Hausdorff and PSO-based algorithm are compared. TABLE I lists the simulation parameters in detail with the number of nodes in the network ranging from 300 to 600. Fig. 8 compares the network lifetime until the first node dies. It is obvious that the algorithm

TABLE I
SIMULATION PARAMETERS

Type	Parameter	Value
Network	Area (m^2)	100 × 100
	Location of data sink	(50, 175)
	Initial energy	2J
	N	From 300 to 600
Application	Data packet length (bit)	100
	Broadcast packet length (bit)	25
Radio model	E_{elec}	50 nJ/bit
	E_{fs}	10 pJ/bit/m^2
	E_{mp}	0.0013 pJ/bit/m^4
	d_0	75m
	E_{DA}	5 nJ/bit/signal

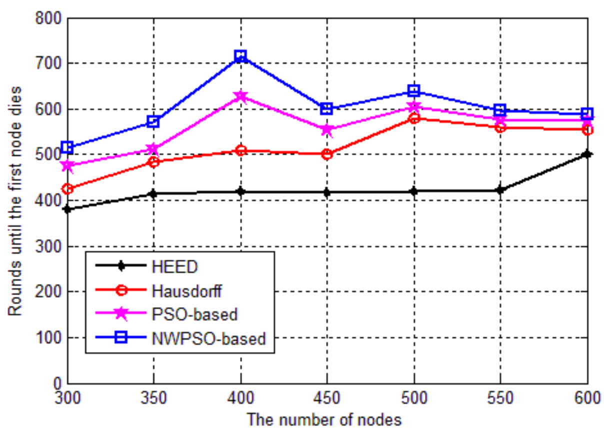

Fig. 8. Lifetime comparison until the first node dies.

is uniformly independent of the node density, and the lifetime of the network has been well prolonged.

The algorithm considers both the nodes' residual energy and minimized transmission distances the same as in HEED and Hausdorff clustering. The algorithm produces a better clustering structure of the network, and the control nodes are distributed more uniformly in the network. The energy consumption of all the nodes is reduced because of a shorter distance between the common nodes and their control nodes. Furthermore, the proposed algorithm uses NWPSO to improve the search of PSO. As a result, the lifetime is prolonged compared with the PSO-based algorithm.

VI. Conclusion

In this paper, we presented a new energy-efficient routing algorithm for the software-defined wireless sensor networks. In our routing algorithm, the control nodes are assigned different tasks dynamically. Meanwhile, we utilize non-linear weight particle swarm optimization algorithm to create a cluster structure so as to minimize the transmission distance and to optimize the energy consumption of the network. Simulation results suggest that the proposed protocol is capable of prolonging the network lifetime.

References

[1] M. Chen, Y. Zhang, Y. Li, M. M. Hassan, and A. Alamri, "AIWAC: Affective interaction through wearable computing and cloud technology," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 20–27, Feb. 2015.

[2] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generat. Comput. Syst.*, 2016, doi: 10.1016/j.future.2015.12.001.

[3] Y. Zhang, M. Chen, S. Mao, L. Hu, and V. C. M. Leung, "CAP: Community activity prediction based on big data analysis," *IEEE Netw.*, vol. 28, no. 4, pp. 52–57, Jul./Aug. 2014.

[4] Y. Zhang, M. Qiu, C. W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Syst. J.*, 2015, doi: 10.1109/JSYST.2015.2460747.

[5] G. Wang, J.-K. Zhang, M. G. Amin, and K. M. Wong, "Nested cooperative encoding protocol for wireless networks with high energy efficiency," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 521–531, Feb. 2008.

[6] A. de la Piedra, F. Benitez-Capistros, F. Dominguez, and A. Touhafi, "Wireless sensor networks for environmental research: A survey on limitations and challenges," in *Proc. IEEE Int. Conf. EUROCON*, Jul. 2013, pp. 267–274.

[7] D. Zeng, P. Li, S. Guo, and T. Miyazaki, "Minimum-energy reprogramming with guaranteed quality-of-sensing in software-defined sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 288–293.

[8] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: Challenges and research advances," *IEEE Netw.*, vol. 28, no. 6, pp. 6–11, Nov. 2014.

[9] X. Ge, B. Yang, J. Ye, G. Mao, C.-X. Wang, and T. Han, "Spatial spectrum and energy efficiency of random cellular networks," *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 1019–1030, Mar. 2015.

[10] M. Chen, Y. Zhang, Y. Li, S. Mao, and V. Leung, "EMC: Emotion-aware mobile cloud computing in 5G," *IEEE Netw.*, vol. 29, no. 2, pp. 32–38, Mar. 2015.

[11] M. Chen, Y. Zhang, L. Hu, T. Taleb, and Z. Sheng, "Cloud-based wireless network: Virtualized, reconfigurable, smart wireless network to enable 5G technologies," *Mobile Netw. Appl.*, vol. 20, no. 6, pp. 704–712, Dec. 2015.

[12] H. Harada *et al.*, "A software defined cognitive radio system: Cognitive wireless cloud," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Washington, DC, USA, Nov. 2007, pp. 294–299.

[13] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *Commun. Lett.*, vol. 16, no. 11, pp. 1896–1899, Nov. 2012.

[14] T. Miyazaki *et al.*, "A software defined wireless sensor network," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Honolulu, HI, USA, Feb. 2014, pp. 847–852.

[15] D. Zhang, G. Li, K. Zheng, and X. Ming, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 766–773, Feb. 2014.

[16] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless charging technologies: Fundamentals, standards, and network applications," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1413–1452, 2nd Quart., 2015.

[17] J. Aweya, "Technique for differential timing transfer over packet networks," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 325–336, Feb. 2013.

[18] L. Militano, M. Erdelj, A. Molinaro, N. Mitton, and A. Iera, "Recharging versus replacing sensor nodes using mobile robots for network maintenance," *Telecommun. Syst.*, vol. 63, no. 9, pp. 1–18, Feb. 2016.

[19] I. Farris, L. Militano, A. Iera, A. Molinaro, and S. C. Spinella, "Tag-based cooperative data gathering and energy recharging in wide area RFID sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 214–228, Jan. 2016.

[20] M. Amjad, M. Sharif, M. K. Afzal, and S. W. Kim, "TinyOS-new trends, comparative views, and supported sensing applications: A review," *IEEE Sensors J.*, vol. 16, no. 9, pp. 2865–2889, May 2016.

[21] A. Lecointre, D. Dragomirecu, and R. Plana, "Software defined radio layer for IR-UWB systems in wireless sensor network context," in *Proc. IEEE Northeast Workshop Circuits Syst. (NEWCAS)*, Montreal, QC, Canada, Aug. 2007, pp. 1149–1152.

[22] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, "SYNAPSE++: Code dissemination in wireless sensor networks using fountain codes," *IEEE Trans. Mobile Comput.*, vol. 9, no. 12, pp. 1749–1765, Dec. 2010.

[23] B. T. de Oliveira, C. B. Margi, and L. B. Gabriel, "TinySDN: Enabling multiple controllers for software-defined wireless sensor networks," in *Proc. IEEE Latin-America Conf. Commun. (LATINCOM)*, Cartagena, Colombia, Nov. 2014, pp. 1–6.

[24] R. Huang, X. Chu, J. Zhang, and Y. H. Hu, "Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype," *Int. J. Distrib. Sensor Netw.*, vol. 2015, Aug. 2015, Art. no. 360428.

[25] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," Internet Eng. Task Force (IETF), Tech. Rep. RFC 3561, Jul. 2003. [Online]. Available: http://www.rfc-editor.org/info/rfc3561

[26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[27] L. Chen and B.-H. Zhao, "Data fusion oriented routing protocol based on leach," *J. Beijing Univ. Posts Telecommun.*, vol. 32, no. 5, pp. 71–74, Oct. 2009.

[28] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct./Dec. 2004.

[29] X. Zhu, L. Shen, and T.-S. P. Yum, "Hausdorff clustering and minimum energy routing for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 990–997, Feb. 2009.

[30] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wireless Netw.*, vol. 15, no. 2, pp. 193–207, Feb. 2009.

[31] L. Mottola and G. P. Picco, "MUSTER: Adaptive energy-aware multi-sink routing in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 12, pp. 1694–1709, Dec. 2011.

[32] Y. Liao, H. Qi, and W. Li, "Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1498–1506, May 2013.

[33] A. Thakkar and K. Kotecha, "Cluster head election for energy and delay constraint applications of wireless sensor network," *IEEE Sensors J.*, vol. 14, no. 8, pp. 2658–2664, Aug. 2014.

[34] D. C. Hoang, P. Yadav, R. Kumar, and S. K. Panda, "A robust harmony search algorithm based clustering protocol for wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Cape Town, South Africa, May 2010, pp. 1–5.

[35] K. B. Lee and J. H. Kim, "Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 755–766, Dec. 2013.

[36] Y. J. Gong, J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung, and Y. H. Shi, "Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 254–267, Mar. 2012.

[37] J. Kennedy, "The particle swarm: Social adaptation of knowledge," in *Proc. IEEE Int. Conf. Evol. Comput.*, Indianapolis, IN, USA, Apr. 1997, pp. 303–308.

[38] R. C. Luo and O. Chen, "Mobile sensor node deployment and asynchronous power management for wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 59, no. 5, pp. 2377–2385, May 2012.

[39] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Commun. Lett.*, vol. 9, no. 11, pp. 976–978, Nov. 2005.

[40] T. Miyazaki, D. Shitara, Y. Endo, Y. Tanno, H. Igari, and R. Kawano, "Die-hard sensor network: Robust wireless sensor network dedicated to disaster monitoring," in *Proc. 5th Int. Conf. Ubiquitous Inf. Manage. Commun. (ICUIMC)*, New York, NY, USA: ACM, Feb. 2011, p. 53.

Wei Xiang (S'00–M'04–SM'10) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 1997 and 2000, respectively, and the Ph.D. degree in telecommunications engineering from the University of South Australia, Adelaide, Australia, in 2004.

He was a Visiting Scholar with Nanyang Technological University, Singapore, in 2008. From 2010 to 2011, he was a Visiting Scholar with the University of Mississippi, Oxford, MS, USA. From 2012 to 2013, he was an Endeavour Visiting Associate Professor with the University of Hong Kong. He has authored over 160 papers in peer-reviewed journal and conference papers. His research interests are in the broad area of communications and information theory, particularly coding and signal processing for multimedia communications systems. He is currently a Foundation Professor and the Head of Discipline Electronic Systems and Internet of Things Engineering, College of Science and Engineering, James Cook University, Cairns, Australia. From 2004 to 2015, he was with the School of Mechanical and Electrical Engineering, University of Southern Queensland, Toowoomba, Australia. He is an IET Fellow, a Fellow of Engineers Australia, and an Editor of the IEEE COMMUNICATIONS LETTERS. He was a co-recipient of three Best Paper Awards at 2015 WCSP, 2011 IEEE WCNC, and 2009 ICWMC. He has been awarded several prestigious fellowship titles. He was named a Queensland International Fellow (2010-2011) by the Queensland Government of Australia, an Endeavour Research Fellow (2012-2013) by the Commonwealth Government of Australia, a Smart Futures Fellow (2012-2015) by the Queensland Government of Australia, and a JSPS Invitational Fellow jointly by the Australian Academy of Science and Japanese Society for Promotion of Science (2014-2015).

Ning Wang received the B.S. degree in communication engineering from Hebei University of Technology, Tianjin, China, in 2014. He is currently pursuing the M.S. degree with the School of Electronic Information Engineering, Tianjin University, Tianjin.

His research interests include protocols design in wireless sensor network and underwater sensor networks.

Yuan Zhou received the B.Eng., M.Eng., and Ph.D. degrees from Tianjin University, Tianjin, China, in 2006, 2008, and 2011, respectively, all in electronic engineering and communication engineering. From 2013 to 2014, she was a Visiting Scholar with the School of Mechanical and Electrical Engineering, University of Southern Queensland, Toowoomba, Australia. Since 2011, she has been a Faculty Member with the School of Electronic Information Engineering, Tianjin University, Tianjin, China, where she is currently an Associate Professor.

Her current research interests include wireless sensor networks, computer vision, and image/video communications.