IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 18, 2018, accepted June 8, 2018, date of publication June 13, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2847036

An Unequal Clustering Algorithm Concerned With Time-Delay for Internet of Things

XIN FENG, JING ZHANG, CHENGHAO REN, AND TINGTING GUAN

College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China

Corresponding author: Jing Zhang (zhang_jing@cust.edu.cn)

This work was supported by the Youth Foundation of Changchun University of Science and Technology under Grant XQNJJ-2017-13.

ABSTRACT Internet of Things (IoT) enables the devices to exchange data with each other. The wireless sensor network is a key technology for making devices sensible and has been widely concerned. In the clustering routing protocol of wireless sensor networks, the cluster heads have high energy consumption rate since it undertakes data collection, fusion, and forwarding, which causes unbalanced energy consumption. Thus, the network lifetime is limited. In this paper, we present the contribution as follows. First, we propose an improved K-means algorithm to cluster the network and use the weighted evaluation function to optimize the cluster structure. Then, we select to either split or merge the cluster structure according to the evaluation results and in further to obtain a non-uniform clustering structure of the network. Second, in the data transmission phase, the data fusion mechanism is used to improve the energy utilization rate of cluster heads. Given the transmission delay problem caused by the data fusion, we propose a delay-optimized data fusion tree construction-based algorithm. When an active node selects the parent node, the distance and the energy factors are considered. The time slot allocation is optimized through constructing a data fusion tree, and the transmission delay is minimized. Finally, compared with other algorithms in the simulation section, the proposed algorithm can effectively reduce the energy consumption of the network, and the constructing data fusion tree decreases the transmission delay caused by the data fusion process. The service quality of the whole network is therefore improved. The proposed algorithm is suitable for the delay-constraint application of IoT.

INDEX TERMS Internet of Things, wireless sensor networks, unequal clustering, K-means, time-delay.

I. INTRODUCTION

With the development of cloud computing [1], [2], big data [3], [4] and mobile Internet [5], Internet of Things (IoT) has been widely applied to fields such as industry [6], medical treatment [7] and smart life [8], [9]. Concerning the Internet of things (IoT), wireless sensor networks (WSNs) become more and more critical in undertaking data monitoring and data transmission [10]. Since the sensor nodes consisting a network have insufficient energy, it is difficult to replenish their energy after the deployment. Therefore, how to utilize the limited energy to maximize the network lifetime has become into focus. Two factors are restricting the network lifetime. First, the sensor nodes collect data and send them periodically to the base station (BS), and the cyclical data transmission will result in a significant amount of data redundancy, which would make the node consume a lot of energy during transmission. Second, since the nodes near the base station would prematurely exhaust their energy due to overloaded forwarding tasks, they might destroy the connectivity of the network and then lead to an "empty hole" problem [11]. To avoid or eliminate the mentioned empty holes led in by the vulnerable network structure, the clustering structure is introduced in WSNs. According to the different tasks of nodes in the network, the network nodes are divided into the cluster heads (CH) and the cluster members (CM). After the data in a cluster is merged by the CH, it is sent to the BS. The method could efficiently reduce the amount of data transferred and the transmission energy consumption. A reasonable CH election strategy can shorten the frequency of cluster reconstruction [12], reduce the energy consumption of cluster formation, and prolong the network lifetime. Data redundancy reduction can also be achieved through data fusion process. However, the construction of data fusion tree and data scheduling can still result in transmission delay and cause the drop of the network service quality. For example, in a smart medical application, when

the patient's condition suddenly deteriorates, if the sensor fails to transmit the patient information to the control center in time, the patient may miss the optimal time of obtaining some treatment, which may even endanger the patient's life. Therefore, how to reduce the delay caused by data fusion is one of the important research work of WSNs.

Based on the above analysis, this paper presents an unequal clustering algorithm concerned with time-delay (UCATD) for IoT. An improved K-Means algorithm is brought into to construct unequal clustering in the aim of solving the ''empty hole'' problem and balance the energy consumption of nodes. The transmission delay caused by data fusion is mainly affected by the data fusion tree structure and data scheduling [13]. Therefore, this paper also provides a data fusion tree with time-delay optimization, which could optimize the delay of the transmission path and the scheduling time slot. Meanwhile, the network lifetime could be prolonged supported by the reduction of the time delay. The main contributions of this paper include:

(1) We propose an improved K-means algorithm to act a non-uniform clustering onto the network and build up a weighted evaluation function of determining whether to merge or split the given cluster. In this way, the cluster structure could be optimized.

(2) During the process of constructing the data fusion tree with the minimum delay, the node level is set to ensure that the nodes at the same level in each time slot can select the corresponding parent nodes to maximize the time slot reuse.

(3) Considering the two factors affecting the network lifetime, i.e., the distance between two nodes and the remaining energy during the process of parent node selection, we choose the optimal nodes to construct the optimal fusion tree so that the final network structure will manifest itself with the minimum delay.

The follow-up paper is organized as follows: Several studies on clustering-based routing algorithms and data fusion algorithms are reviewed in Section 2. The models used in this study are described in Section 3. Section 4 presents the unequal clustering algorithm implementation process and a delay optimized data fusion tree construction process. A series of experiments are presented in Section 5. The conclusion is drawn from the research results in Section 6.

II. RELATED WORKS

The problem of unbalanced energy consumption in the clustering routing protocol of WSNs is a hot research topic in recent years. Heinzelman *et al.* [14] proposed the LEACH. The periodic CHs election is made during network operation, and then the node energy consumption is balanced. A data fusion mechanism is also used to reduce the data transmission energy consumption. Reference [15] was proposed to reduce the probability of forming energy hole near the base station by increasing the node density near the base station in node deployment. The algorithm is based on the clustering strategy used in LEACH protocol, in line with the needs of balancing network load, but not applicable for the network with node random spreading. Reference [16] proposed a density-based energy-efficient game theory routing algorithm. The algorithm sets the utility function according to the node density, residual energy and average energy consumption of neighbor nodes. The iterative method is used to replace the CHs and the data transmission adopts intra-cluster and inter-cluster multi-hop routing algorithm. The algorithm effectively uses a variety of parameters for the CH election, but the parameter calculation process is relatively complicated, which will increase the overall network transmission delay. The EEUC algorithm proposed in [17] is an early non-uniform routing protocol. This protocol preliminarily solves the problem that the nodes that are closer to the base station die too soon by making the clusters closer to the base station smaller than the clusters far away from the base station. However, the procedure of determining the clustering radius in EEUC algorithm only depends on few factors and its proposed clustering radius would be unreasonable. In [18], UCR algorithm is proposed as a non-uniform clustering algorithm. However, during the operation of the algorithm, the node's competition radius is unchanged and its proposed energy consumption would thus be uneven. The CUCRA algorithm [19] is also a non-uniform clustering algorithm. But the energy factor is taken into account when calculating the competition radius, so that the node's competition radius becomes smaller as the node's residual energy is less. The algorithm proposed in the literature [20] introduces the DB evaluation function to obtain the number of clusters, and it uses the Gaussian evaluation to measure the effectiveness of the cluster head, which could reduce the energy consumed during each round of cluster head replacement. The BPK-means algorithm proposed in [21] adopts a balanced scheduling strategy after cluster clustering so that the number of nodes in each cluster tends to average. In this way, the total energy consumption in each cluster is equalized. Since the communication phase according to the algorithm employs the single-hop mode, the algorithm is not proper to run under the large-scale network model. Based on the k-means algorithm clustering, the EKMT algorithm proposed in [22] integrates the distance from the node to the cluster center point and the distance from the node to the base station as the decision elements in the cluster head election strategy. The algorithm takes the remaining energy factor and could satisfy the energy balancing requirements, but it could still not solve the energy consumption hotspot problem in the many-to-one transmission mode. However, once the nodes with more remaining energy are too concentrated, the CH distribution will be unreasonable. In clustering routing protocol, the CH conducts data fusion to enhance its energy utilization. But the above algorithms do not consider the transmission delay problem during data fusion process.

Time-delay is an important criterion to measure the quality of service (QoS) of WSNs. It usually refers to the total delay required to transmit one (or a group) of data packets from the source node to the destination node, including the propagation delay, queuing delay and routing delay, etc. Data fusion

can reduce the redundancy of the data, achieve the purpose of saving the network energy and improving the data reliability, but it inevitably causes the transmission delay and makes the QoS guarantee of WSN more complicated. Especially concerning the applications that require the transmission of real-time data (such as images or video), it becomes more urgent to focus on the transmission delay. PEGASIS (power-efficient gathering in sensor information systems) [23] is a nearly optimal chain-based protocol originated from LEACH, which connects all the nodes in the network into a link in which the adjacent nodes have the shortest distance. Through randomly selecting a node as the leader from both ends of the link, the nodes, in turn, send data to the leader node, and the intermediate nodes will conduct fusion processing of the received data, and send the fusion results to the next node. Eventually, the data is sent to the BS by the leader node. The disadvantage of PEGASIS is that it takes some additional resources for nodes to maintain their location information. The algorithm NCA [24] (Nearly Constant Approximation for Data Aggregation Scheduling) establishes a Connected Dominance Set (CDS) as a fusion tree. The dominating node is firstly scheduled, and then the nodes in the CDS are dispatched progressively from the bottom up using the first fit time slot assignment process. [25] proposed a centralized and improved data fusion scheduling algorithm (CIAS) as an improved algorithm based on NCA. By building a data fusion tree routed to the center of the network, the network center receives the fusion data and forwards the fusion result to the BS. In [26], a heterogeneous RBF neural network information fusion algorithm is proposed, which is used to converge the heterogeneous information of aggregation nodes with good real-time performance and small network delay. Besides, the algorithm can reduce network conflicts and congestion. In [27], a centralized algorithm was proposed, in which no information about the candidate parent nodes or the child nodes is provided to the scheduling algorithm. Tree establishment and scheduling are carried out simultaneously. According to [28], DADCNS algorithm reduces the data fusion delay by constructing a network structure with delay optimization, but it does not adequately control the energy to save and the energy balance among nodes, which shortens the network lifetime. The above fusion methods are all based on a particular network structure. Reference [29] proposed an extensible unstructured data fusion algorithm (SP), which is a distributed dynamic fusion algorithm. The nodes merge the data at the end of waiting time and send the merged data to the best neighbor chosen by the proposed best neighbor algorithm. The algorithm is superior to other algorithms regarding scalability and convergence. The SFEB (structure-free and energy-balanced data aggregation protocol) proposed in [30] is also an unstructured and energy-balanced fusion algorithm. Through the two stages of the fusion process and the dynamic selection mechanism deployed within the fusion machine, it could achieve efficient data collection and energy consumption balance. However, in general, the energy consumption performance of unstructured fusion is not as good as the energy consumption performance brought by the structure algorithm. Therefore, this paper will adopt a data fusion method based on network structure.

To ensure the maximum network service quality, we mainly focus the balanced energy consumption and the transmission delay as the key factors. We thus propose an unequal clustering algorithm to cope with ''energy hole'' and the delay problem of data fusion. This paper firstly designs a uniform energy-concentration clustering strategy based on the improved k-means clustering algorithm. The introduced clustering algorithm is used to cluster the network nodes reasonably to avoid energy distribution imbalance caused by unreasonable clustering. Then, referring to the hot-spot problem caused by the multiple-to-one transmission mode, a splitting and merging strategy is proposed, which could thus reduce the energy consumption in the area close to the base station and extend the network lifetime. Second, we evaluate the factors that affect the data transmission delay and build a delayed optimized data fusion tree, which maximizes the time slot utilization and reduces the transmission delay. The proposed algorithm is optimized regarding energy consumption and transmission delay.

III. MODELS PRESENTATION

A. NETWORK MODEL

The network model assumption used in this paper is as follows:

(1) N sensor nodes are randomly deployed within the monitoring area with an area of $S = M^*M$. The BS is located at the center of the network, and both the sensor nodes and the base stations are stationary.

(2) Nodes in the network have the same initial energy, and each node has a unique ID identifier. The nodes have its limited energy, and the BS has infinite energy.

(3) The link is symmetric. The node can calculate the approximate distance between the sender and itself based on the received signal strength.

(4) Each node only needs to spend one time-slot to communicate with its parent node, and each node can only receive or send one data packet and corresponding control packet in this time slot.

(5) The node can adjust the transmission power according to the communication distance.

B. ENERGY CONSUMPTION MODEL

Most of the energy consumption of sensor nodes is spent in the data communication. Thus, we only assume the energy consumption cost in the data transmission and the fusion data in this paper. The following equations formalize the energy consumption of transmitting and receiving.

$$E_{TX}(k,d) = \begin{cases} kE_{elec} + k\varepsilon_{fs}d^2, & d < d_0 \\ kE_{elec} + k\varepsilon_{amp}d^4, & d \geq d_0 \end{cases} \quad (1)$$

$$E_{RX}(k) = kE_{elec} \quad (2)$$

where, k represents the data length, i.e., the number of bits. d represents the data transmission distance, and E_{elec} represents

the energy consumed during the transmission and reception of the unit length data. ε_{fs} and ε_{amp} represents the amplifier power consumption of the free-space model and multi-path attenuation model respectively. When the distance d between the transmitting node and the receiving node is less than the energy consumption model threshold d_0, the free space model is adopted and the transmission power is attenuated as d^2. Otherwise, the multi-path attenuation model is adopted and the transmission power is specified as d^4.

The energy consumption required for nodes to fuse k-length data is formalized as:

$$E_{\mathrm{A}}(k) = kE_{DA} \tag{3}$$

where E_{DA} is the energy consumption required to fuse a unit length of data.

IV. THE DETAIL OF UCATD

Clustering routing algorithms execute by rounds, and each round includes two phases, namely cluster formation and data transmission. In the clustering stage, the number of optimal clusters is first calculated, and the initial clustering centers required by the K-Means clustering algorithm are selected by using the region division method, and clustering is performed using the objective function. To avoid the uneven distribution of node load caused by the many-to-one transmission mode, this paper adopts the splitting and merging operations to adjust the scale of the region with large energy consumption so as to balance the energy consumption of nodes in the network. In aim of ensuring the cluster head election validity, this paper suggests a weighted evaluation function to select the optimal cluster head in consideration of position and energy, and to improve the node energy balance in the network. In the data transmission phase, a delay optimized data fusion tree is constructed to reduce the influence of data fusion on transmission delay.

A. CLUSTER FORMATION

1) DETERMINATION OF OPTIMAL CLUSTER NUMBER

The rational determination of the cluster amount in WSN is usually based on the consideration of energy efficiency. If there are too many clusters, too much clustering cost would be paid; if the number of clusters is too small, it would lead to too many nodes in each cluster and a number of cluster heads consuming more energy would die too soon. So a reasonable number of clusters can not only effectively improve the efficiency of the network link, but it could also balance the node energy loss and extend the network life cycle.

In this paper, the inter-cluster communication employs the multi-hop routing mode. The distance from the base station to the furthest cluster head is set to be D. This distance is divided into multiple hops. For the convenience of discussion, we utilize the linear equidistant model as shown in Figure 1.

As illustrated in Figure 1, $D = k \cdot d$, k is the number of cluster heads, and d is the length of equidistance. Under

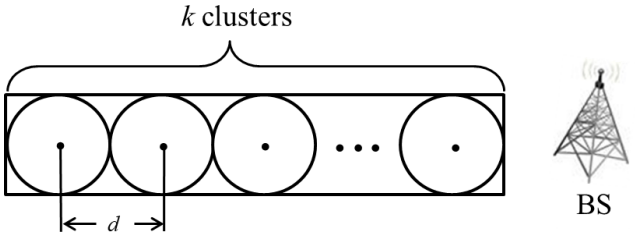

FIGURE 1. Multi-hop equidistant model.

the multi-hop transmission model, the energy consumption is expressed as:

$$E_{multihop} = E_{Rx} + E_{DA} + E_{Tx} \tag{4}$$

If $d < d_0$, c is the data compression ratio or data fusion ratio (i.e., the data quantity before the compression/fusion is divided by that after the compression/fusion). Then,

$$\begin{aligned} &E_{multihop} \\ &= (E_{elec} \cdot l + \varepsilon_{fs} \cdot l \cdot d^2)_1 \\ &\quad + (E_{elec} \cdot l + E_{da} \cdot l + E_{elec} \cdot c \cdot l + \varepsilon_{fs} \cdot c \cdot l \cdot d^2)_2 \\ &\quad + \left(E_{elec} \cdot c \cdot l + E_{da} \cdot c \cdot l + E_{elec} \cdot c^2 \cdot l + \varepsilon_{fs} \cdot c^2 \cdot l \cdot d^2\right)_3 \\ &\quad + \ldots + (E_{elec} \cdot c^{k-2} \cdot l + E_{da} \cdot c^{k-2} \cdot l + E_{elec} \cdot c^{k-1} \cdot l \\ &\quad + \varepsilon_{fs} \cdot c^{k-1} \cdot l \cdot d^2)_k \end{aligned} \tag{5}$$

when $c = 1$,

$$\begin{aligned} E_{multihop} = E_{elec} \cdot l \cdot (2k-1) + E_{da} \cdot l \cdot (k-1) \\ + \varepsilon_{fs} \cdot l \cdot d^2 \cdot k \end{aligned} \tag{6}$$

The total energy consumed through multi-hop transmission could be denoted as the sum of the energy consumption cost among the clusters and the energy consumption cost among the nodes within each of the clusters which can be formulated as (6).

$$E_{total} = E_{multihop} + k \cdot E_{incluster} \tag{7}$$

The energy consumption within the cluster can be derived according to the free-attenuating channel model as demonstrated by (8).

$$E_{incluster} = \left(\frac{N}{k} - 1\right) \cdot l \cdot E_{elec} + \left(\frac{N}{k} - 1\right) \cdot l \cdot \varepsilon_{fs} \cdot d_{toCH}^2 \tag{8}$$

Assume that the network area is fully covered by the k circular cluster domains, then

$$M^2 = \pi \cdot R^2 \cdot k \tag{9}$$

The distance between clusters could be calculated by $d = 2R = \frac{2M}{\sqrt{\pi k}}$ as shown in Figure 2.

$$\begin{aligned} E\left(d_{toCH}^2\right) &= \iint (x^2 + y^2) \cdot \rho(x, y) dxdy \\ &= \rho \cdot \iint (x^2 + y^2) dxdy \end{aligned}$$

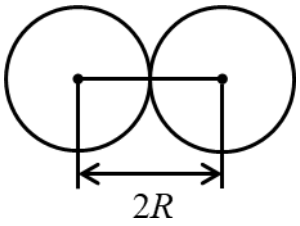

FIGURE 2. Distance between clusters.

$$= \rho \cdot \int_{\theta=0}^{2\pi} \int_{r=0}^{\frac{M}{\sqrt{\pi k}}} r^3 dr d\theta \tag{10}$$

Assume that the distribution of nodes is uniformly distributed, then

$$\rho = \frac{1}{M^2/k} \tag{11}$$

With putting (11) into (10), we could obtain:

$$E\left(d_{toCH}^2\right) = \frac{M^2}{2\pi k} \tag{12}$$

Thus, E_{total} is expressed as (13):

$$E_{total} = (2k-1)\cdot E_{elec}\cdot l + (k-1)\cdot l \cdot E_A + 4M^2 \cdot \varepsilon_{fs} \cdot \frac{l}{\pi} + N \cdot l \cdot E_{elec} + N \cdot l \cdot \varepsilon_{fs} \cdot \frac{M^2}{2\pi k} \tag{13}$$

In order to calculate the k-value that minimizes the total energy consumption, we calculate the derivative of E_{total} with k and set the derivative as zero in order to induce the optimal number of clusters k_{opt} to minimize the total energy consumption of the network.

$$\frac{\partial E_{total}}{\partial k} = 2E_{elec} \cdot l + E_{da} \cdot l - N \cdot l \cdot \varepsilon_{fs} \cdot \frac{M^2}{2\pi} \cdot \frac{1}{k^2} \tag{14}$$

Set $\frac{\partial E_{total}}{\partial k} = 0$, then

$$k_{opt} = \sqrt{\frac{N \cdot \varepsilon_{fs} \cdot M^2}{2\pi \cdot (2E_{elec} + E_A)}} \tag{15}$$

2) STAGE OF FORMING CLUSTERS

When using K-means algorithm to cluster data, the choice of the initial cluster center would directly affect the clustering result and may have a great impact on the performance of clustering. The K-means algorithm is a local search clustering algorithm. The result of the algorithm depends on the initial state of the process, i.e., the selection of the initial cluster center point, and the algorithm can only guarantee convergence to a fixed point and cannot guarantee convergence to the minimum point of the objective function. The algorithm sometimes might converge to the saddle point of the objective function. Therefore, to use K-means algorithm for obtaining an optimized clustering structure, reasonably determining the initial clustering center is an important step during clustering implementation. The basic idea and procedures of the cluster center point selection proposed in this paper are as follows.

To determine the distribution of nodes in an area, the centroid of the nodes in the specified area $G_k(\overline{x}, \overline{y})$ is found by (16).

$$\overline{x} = \frac{\sum_{i=0}^{n} x_i}{n}, \quad \overline{y} = \frac{\sum_{i=0}^{n} y_i}{n} \tag{16}$$

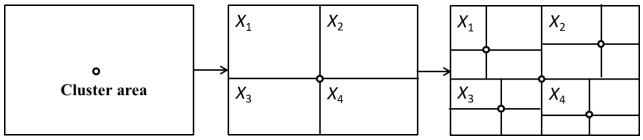

FIGURE 3. An example of dividing area.

where, n is the number of nodes. Focusing on the centroid of the area, we divide the entire area and get the initial four areas X_1, X_2, X_3, X_4. All nodes record the area information, and the centroid of the new area could be recalculated. If $k > 4$, then the four obtained areas (i.e., X_1, X_2, X_3, X_4) would be divided in the same way to obtain 16 areas. The dividing process is demonstrated as Figure 3. The number of the divided areas a is related to the number of clusters k:

(1) when $k < 4$, the dividing process are executed to obtain 4 areas, i.e., $a = 4$;

(2) when $4 \leq k \leq 16$, a second dividing process is executed to obtained 16 areas, i.e., $a = 16$;

(3) when $k > 16$, a third dividing process is executed to obtain 64 areas, i.e., $a = 64$, and so on.

Count the number of nodes in each area, and use the centroid of the area having the most nodes as the first cluster center p_1. Calculate the distance between the centroid of other area G_a and the first cluster center p_1 in turn and select the point with the largest distance to the first cluster center as the second cluster center, i.e., p_2. Keep calculating the distance (i.e., $d(G_a, p_1)$, $d(G_a, p_2)$) from the centroids of the other areas to the determined cluster centers. Select the centroid of the area which has $\max[d(G_a, p_1) + d(G_a, p_2)]$ as the third cluster center and so on. Then the k-th cluster p_k could be induced according to (17).

$$p_k = \max\left(\sum_{i=1}^{k-1} d(G_a, p_i)\right) \tag{17}$$

The specific clustering steps are as follows:

Step 1: put the obtained k cluster centers into the equation.

Step 2: calculate the distances from the n nodes to the cluster center point of k clusters. Each node selects to join the cluster with the shortest distance.

Step 3: calculate the geometric mean of the nodes in each cluster as a new cluster center point.

Step 4: utilize the error square sum criterion (i.e., denoted as (18)) to determine if the error criteria have been reached. If not, return to Step 2 to continue. Otherwise, the clustering

process ends and k classes are taken as the output.

$$J = \sum_{i=1}^{k} \sum\nolimits_{x \in C_i} (\|x - \mu_i\|^2) \tag{18}$$

3) CLUSTER STRUCTURE OPTIMIZATION

a: DETERMINING THE NUMBER OF NODES IN THE OPTIMAL CLUSTER

In a WSN, the many-to-one data stream transmission mode makes the cluster heads that is closer to the base station carry out the more data volume, which leads the cluster heads consume energy faster. In this paper, the competition radius [17] is introduced to describe the relationship between the number of node in a cluster and the distance to the BS. It is expressed as:

$$R_i = (1 - \tau \cdot \frac{d_{\max} - d_{(S_i,BS)}}{d_{\max} - d_{\min}}) \cdot R_0 \tag{19}$$

The competition radius of the node defined by (19) is the initial competition radius, in which τ is the factor that adjusts the scope of the competition radius and determines the impact of the distance on the competition radius. The greater the τ is, the greater the impact of distance on the competition radius. R_0 is the maximum competition radius. $d_{\max}$ and $d_{\min}$ are the maximum and minimum distances from all nodes to the base station. When τ increases, the variation ranges of R_i value decreases; conversely, when τ decreases, the range of variation of R_i value increases, and R_0 directly affects the value of R_i. From Equation 19, we can conclude that the competition radius of the cluster is proportional to the distance from the cluster to the base station. The competition radius of the cluster always varies between R_0 and $(1-\tau)\,R_0$, The closer a cluster is to the base station, the smaller the competition radius is. The smaller the competition radius is, the less energy is used to manage the members of the cluster, so that it has more energy for data forwarding during multi-hop transmission communications.

Assume that the nodes are randomly distributed within the two-dimensional plane with satisfying the uniform distribution, the probability density of the nodes could be obtained, and the cluster radius obtained by combining (19) can obtain the reasonable value of the nodes amount in each cluster. Its expression is expressed as:

$$N_i = \pi \cdot R_i^2 \cdot \rho \quad (i = 1, \cdots, k) \tag{20}$$

b: SPLITTING AND MERGING

According to the core idea of the K-means clustering algorithm, the clustering effect of the algorithm is reflected in the grouping of nodes close to each other into one cluster. The size of the cluster domain is inhomogeneous and will cause the problem of ''energy hole'', i.e., the clusters close to the base station has too many nodes, which consumes a large amount of data and the more energy. Especially, due to the long distance, the clusters will make some uneven energy consumption and then affect the function of the entire WSN. Therefore, this paper proposes a split-and-merge operation based on energy balance to adjust the cluster domain derived from the K-means algorithm. It does not use the repeated iterative method for splitting and merging which could preserve the best clustering effect of the K-means algorithm.

When selecting clusters to be adjusted, a weighted evaluation function is proposed whose expression is:

$$W(i) = \alpha \cdot \frac{D_i - D_c}{D_{\max} - D_{\min}} + \beta \cdot F(i) \tag{21}$$

$$F(i) = \begin{cases} \dfrac{(1+c) \cdot N_i - n_i}{n_i - N_i} (n_i > (1+c) \cdot N_i) \\ \dfrac{(1-c) \cdot N_i - n_i}{N_i - n_i} (n_i < (1-c) \cdot N_i) \end{cases} \tag{22}$$

The weight function in equation (21) considers the distance from the cluster to the base station and the number of nodes in the cluster. In the equation, D_i is the distance from the cluster S_i to the base station. D_c is the average distance from the center point of all clusters to the base station, $D_{\max}$ and $D_{\min}$ represent the maximum and the minimum distances from the center point of all clusters to the base station. The denominator of the equation is $D_{\max}$-$D_{\min}$. With the denominator, the value of the first part can be made within 0 to 1, effectively playing a normalizing role. $F(i)$ means the influence of the node amount in the cluster on the evaluation value. α and β denote the influence weight of the distance factor and the number of nodes onto the evaluation value respectively.

In (22), N_i is the number of optimal cluster nodes found in (20). n_i is the actual number of nodes in the cluster after K-means clustering, and c is the reasonable range of the node amount in the cluster. If the value of n_i is in the range of [$(1 - c)^*N_i$, $(1 + c)^*N_i$], it is regarded as a reasonable number of nodes in the cluster and the cluster should not be split or merged.

The specific algorithms for splitting and merging operations are as follows:

Step 1: Splitting operation. Traverse all clusters S_i, and select the specific cluster whose actual node amount $n_i > (1 + c)^*N_i$. Then calculate the weighted evaluation values $W(i)$ using (21) and (22), and sort $W(i)$. Select $W(i)$ from the clusters whose $W(i)$ is less than 0 from small to large $W(i)$;

Step 2: Calculate the standard deviation of the nodes, i.e., S_i, in the cluster that need to be split. Split the cluster average into two cluster blocks whose centers are c_i^+ and c_i^- and discard the original center. Let $k = k + 1$. c_i^+ and c_i^- are calculated as follows: Given an h value such that $0 < \mathrm{h} \leq 1$, $c_i^+ = c_i + h\sigma_i$, $c_i^- = c_i - h\sigma_i$ where the value of h is chosen to make the distance from the point in the cluster S_i to c_i^+ and c_i^- are different, and meanwhile it is necessary to ensure that the previous sample of S_i is still in the two new sets.

Step 3: Merging operation. Traverse all the clusters $\mathrm{S_i}$ and use (21, 22) to find the weighted evaluation values $W(i)$. Then sort $W(i)$, and select from the largest to the smallest $W(i)$ among the clusters whose $W(i) > 0$;

Step 4: Calculate the distance d_{ij} from the center point C_i of clusters S_i that need to be merged to all other cluster

centers. Take the clusters S_j and S_i who commonly have the minimum d_{ij} and merge them. The cluster center after merging is expressed as:

$$C_l = \frac{N_i \cdot C_i + N_j \cdot C_j}{N_i + N_j} \tag{23}$$

Corresponding to the center C_l, the original centers C_i and C_j are then discard. Then the number of cluster centers turns to $k = k - 1$.

4) CLUSTER HEAD ELECTION

After the clusters is determined, the BS will send a broadcast message to the node, and the message includes the cluster ID of the cluster. After receiving the broadcast message, the nodes in each cluster distribute the election for the cluster head.

When considering the distance and energy factors, the weighting evaluation function is as follows:

$$W_{DE}(i) = w \cdot \frac{E_c(i)}{E_{aver}} \cdot (D_{\max} - D_{\min}) + (1 - w) \cdot \frac{D_{aver}}{D(i)} \tag{24}$$

where, E_c and E_{aver} represent the current residual energy of each node in the cluster and the average energy of all nodes in the cluster respectively. $D(i)$ and D_{aver} represent the distance from the given node, i.e., n_i to the cluster center and the average distance from all the nodes to the cluster center. From (24), it can be seen that in a cluster, the more residual energy of a node, the closer the node is to the cluster center and it is firstly selected as a cluster head node.

After completing the CH election, the ordinary nodes begin to select the CH. The CH node broadcasts the elected CH message with the maximum election radius $R_{\max}$, and the nodes within the coverage of CH send its joining message according to the strength of the received signal. The ordinary node selects the cluster to join according to the received CH broadcast message, and the CH uses TDMA strategy to allocate time slots to the CMs. After receiving the assigned time-slots, all CMs send the monitoring data information to the CHs within their assigned time slots.

The flow of the clustering process is presented as figure 4. After deploying the nodes, the optimal cluster amount is first calculated according to the energy consumption model, and the optimized clustering center is determined by using the area division method. Based on the obtained optimal cluster amount and clustering center, the network is clustered using the k-means method. The objective function is equation (18). Relying on the relationship between the node location and the cluster radius described by the competition radius. The cluster structure is then evaluated and the splitting and merging operations to optimize the cluster structure are executed. During the cluster head election stage, the weighting function is constructed based on considering the distance and the node residual energy to balance node energy consumption in the network. The algorithm is executed in turn, and the cluster heads are alternated periodically, thereby to ensure the energy balance of the nodes.

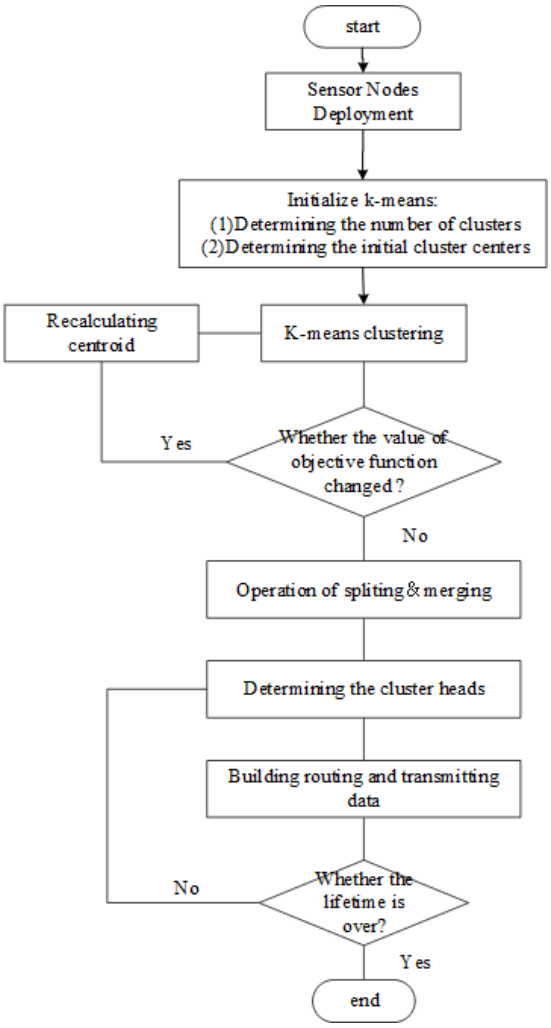

FIGURE 4. The flow of the clustering process.

The clustering process can be divided into two part: the cluster forming and cluster structure optimization. In the cluster forming phase, it is consist of the computation process of the cluster center and K-means iterative process, the time complexity is $O(k^*a)$ and $O(k^*t)$, respectively, where k is the optimal number of clusters, a is the number of initial zones and t is the number of iterative. In the cluster structure optimization phase, the time complexity is $O(k)$.

B. THE PROCESS OF DELAY OPTIMIZED DATA FUSION TREE CONSTRUCTION

In order to reduce data redundancy and save part of transmission energy consumption in network, the CH executes data fusion when data transmission is carried out. Considering the transmission delay caused by data fusion, a data fusion tree is constructed in this paper based on delay optimization, the detail of data fusion tree constructing process consists of the following three steps:

1) INITIALIZATION

At first, all of CHs are independent and do not make any connections with other nodes, so they are all active nodes and have a level of 0. In this algorithm, only the nodes with the same level of the subtree can be connected with each other to form a minimum delay fusion tree. If a connection request comes from a node with different level, the node will reject the connection request and send a refusal message. If a node receives a rejection message, it will send a connection request to its neighbors.

2) BUILDING NODE PAIR

All CHs in the network broadcast their connection requests within their transmission radius. After receiving a connection request, each node can estimate the distance to each adjacent node according to the signal strength. When a node selects a neighbor node for connection, if the neighbor node has already connected to other nodes, the node will select the neighbor node second nearest to it for connection. We treat each node pair as a subtree, so the rank of each node in the subtree is ($\log_2^2 = 1$). If there is still no connection with other nodes within the node coverage, it will be directly connected with the BS. The node directly connected to the BS is not changed in the level value. The level of node i is denoted as

$$\text{Level}_i = \log_2(\text{SUM}_i) \tag{25}$$

where, SUM_i represents the total number of nodes in the subtree where node i is currently located.

If the connection between two nodes in the network that are the closest to each other, the two nodes that are far away from each other need to be connected finally, it will increase energy consumption. To avoid this phenomenon, the function that is inversely proportional to the distance is used to control the order of selecting one node's adjacent nodes. In this way, the nodes farther away from the base station will select the adjacent nodes preferentially, and the remaining nodes will be concentrated near the BS. The overall length of the network structure will be shortened.

$$T_w(d_{i,BS}) = \frac{1}{d_{i,BS}} + random(0, c) \tag{26}$$

The waiting time of node i is calculated by (26). $d_{i,BS}$ denotes the distance from node i to the BS. According to the range of the first part, the value of c is set to 0.001. Random function is introduced into to avoid conflicts between nodes with equal distances from the BS. An example of node-to-node connection is shown in Figure 5.

3) CONSTRUCTING DATA FUSION TREE

After the connection of independent nodes, multiple pairs of connected nodes form in the network. Then the nodes should be connected, and the nodes to be connected are named as active nodes. The two active nodes in each sub-tree exchange their respective adjacent node information and the one with a smaller weight sends the connection request to the other. If the two nodes have the same level, a connection is established

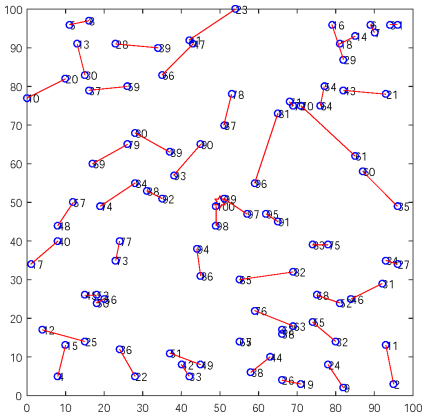

FIGURE 5. Node-to-node connection.

between the two nodes. The two nodes work as the active nodes of the new composition tree. The other nodes keep the current connection, and the algorithm ends. If the two nodes have different levels, it will reject the connection request and send a rejection message. This process is repeated until all the nodes are connected to the data fusion tree.

Although the communication distance between nodes is minimum, there could be other problem. If the distance between nodes is taken as the weight of the edge, the minimum spanning tree is fixed in each round. Because the position of the node is unchanged and the distance between the nodes is fixed. For the nodes with more branches, it will consume in each round a significant amount of energy to receive, fuse, and send data. However, the nodes with fewer branches would consume less energy. To prolong the network service life, we should keep the energy consumption balanced as much as possible. The remaining energy of the node should be examined when it comes to the connection weight. The node with more residual energy is more likely to be selected as the parent. Otherwise, the probability is smaller. When two nodes consume less energy in data transmission and meanwhile the total residual energy is more, the possibility that their connected edges to be selected during the tree spanning is larger so as to balance the energy consumption of all the nodes in the network. When a data transmission is performed between two nodes i, j, the total energy consumed by them can be expressed as follows:

$$E_{\text{COS}} = E_{TX} + E_{RX} = \begin{cases} 2kE_{elec} + k\varepsilon_{amp}d^2, & d < d_0 \\ 2kE_{elec} + k\varepsilon_{fs}d^4, & d \geq d_0 \end{cases} \tag{27}$$

then the energy weight $W(E)_{i,j}$ of the edge can be expressed as:

$$W(E)_{i,j} = \frac{E_{\text{COS}}}{E_i + E_j} \tag{28}$$

where, E_i and E_j are the residual energy of two nodes respectively.

When two subtrees are merged, only the two nodes participating in the join continue to participate in the next round of merge as active nodes of the composite subtree, while

the other nodes will keep the current connection and end the algorithm. In this way, the density of candidate parent nodes in the network becomes smaller that could increase the distance between them. Therefore, treating the weight of the link should also take the distance between the two parent nodes and the BS into account so that the candidate parent nodes of the synthetic subtree converges at the BS. In this way, the distance between the candidate parents becomes shorter, and eventually, the parents of all the sub-trees gather near the BS, so that their communication distance to the BS drops. The distance weight $W(D)_{i,j}$ of the edge to which node i, j is connected can be expressed as:

$$W(D)_{i,j} = \frac{d_{i,j}}{d_{i,BS} + d_{j,BS}} \tag{29}$$

And then the weight function can be expressed as:

$$W_{i,j} = \alpha W(E)_{i,j} + (1 - \alpha) W(D)_{i,j} \tag{30}$$

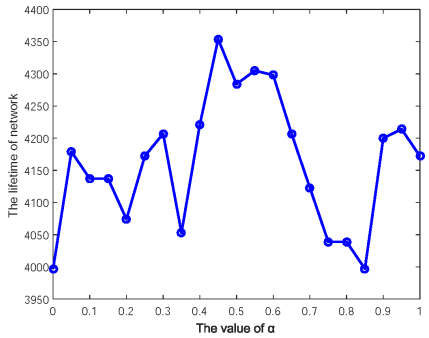

FIGURE 6. The value of α.

where, the coefficient $\alpha \subset [0, 1]$ is a parameter that controls the convergence speed of the candidate node to the BS. If the value of α is larger, the node with larger remaining energy is more advantageous when selecting the adjacent node. Conversely, when the value of α is smaller, the nodes with shorter distances are more advantageous. In order to obtain a better delay result, we must choose an appropriate α value. In this paper, simulation experiments are conducted to determine the value of α. Figure 6 shows the impact of different α values on the network lifetime. Because the data fusion delay is only associated with the formation of the data fusion tree structure and data scheduling, but not related to the value of α. The optimal value of α is chosen through examining the effect of α on the network lifetime. Based on the iterative experiments, the optimal value of α is acquired as 0.45.

The implementation of the algorithm is shown in Figure 7.

C. DELAY ANALYSIS

The transmission delay proposed in this paper refers to the time required for all nodes in the network to collect data in a round. This parameter is particularly crucial for time-limited applications. In the data transmission process, each node is assigned a time slot. Within its time slot, the node begins to transmit data to its parent node. Assuming that

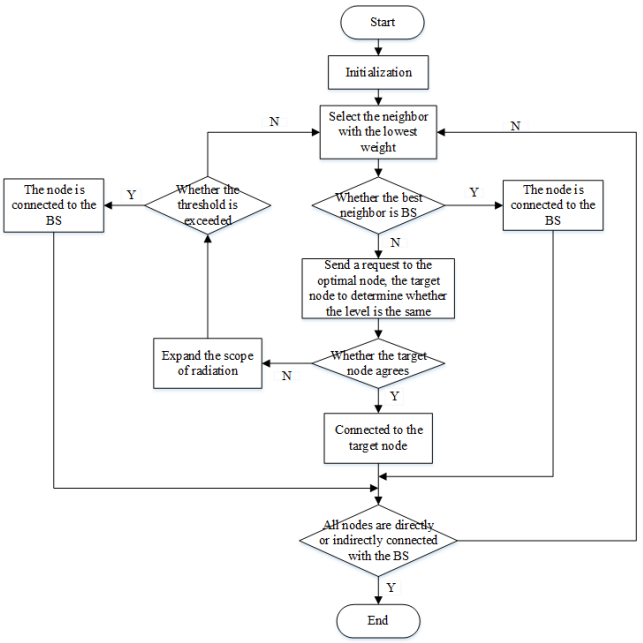

FIGURE 7. Fusion tree construction flow chart.

the time required for any two nodes to transmit a data is t and the total number of time slots allocated in a round is m, the delay caused by this round of data transmission is m^*t. The following is the analysis and comparison of the delay generated by this algorithm and PEGASIS algorithm in each round. Suppose there are n nodes in the network.

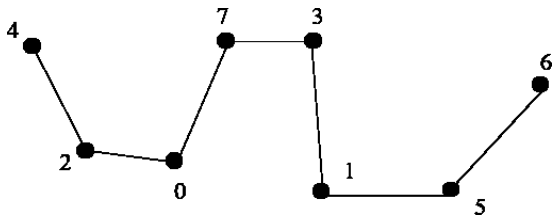

FIGURE 8. PEGASIS link.

PEGASIS eventually generates a linked list, and it selects a root node randomly. The root node sends the fusion data to the BS in each round. As shown in following Figure 8, there are 8 nodes in the network. If node 3 is selected as the root node, the data will be transmitted in the following order: $4 \rightarrow 2 \rightarrow 0 \rightarrow 7 \rightarrow 3 \leftarrow 1 \leftarrow 5 \leftarrow 6$. Because the root node is waiting for receiving data from both ends of the chain, so PEGASIS can generate a significant delay. Especially when the root node is at either end of the linked list (such as node 4 or 6), the data transfer can only start with the node at the other end and only one pair of nodes in per slot can work. Since the linked list has $n - 1$ edges, the total number of the allocated time slots is $n - 1$. Assuming that the time required for transmitting data from the root node to the base station is t', the total time required to transmit the data is $(n - 1)^*t + t'$.

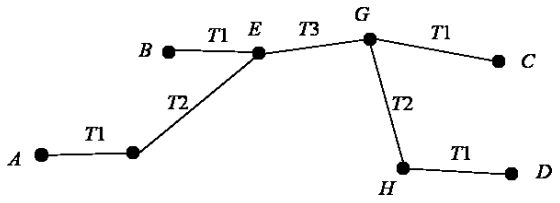

FIGURE 9. Time slot allocation in UCATD.

Referring to the fusion tree constructed in this paper, there are $n/2$ nodes in the first time slot being transmitted at the same time, there are $n/2^2$ nodes in the second time slot, Thus, there are $n/2^m$ nodes in the m-th time slot, then the total number of allocated time slots is $\lceil \log_2 n \rceil$. As shown in Figure 9, the network contains eight nodes. Time slot T1 has 4 pairs of nodes to transmit simultaneously, and time slot T2 has 2 pairs of nodes to transmit simultaneously, and time slot T3 has 1 pairs of nodes to transmit simultaneously. The optimal situation for the network of 8 nodes is to allocate 3 time slots, and the total time required to transmit data is $\lceil \log_2 n \rceil^* t + t'$.

Time slot T1: A → F, B → E, C → G, D → H;
Time slot T2: F → E, H → G;
Time slot T3: E → G.

V. SIMULATIONS

A. SIMULATION PARAMTER SETTINGS

In this paper, we use MATLABR2016a to simulate the proposed DOUCA algorithm and analyze the performance through comparing the proposed algorithm with EKMT, UCR and CUCRA. The experimental environment and necessary parameters are listed in Table 1.

TABLE 1. Parameter settings.

Parameter	Parameter value
Node coverage area	(200m,200m)
BS location	(100m,200m)
Number of nodes	500
E_{elec}	50nJ/bit
e_{fs}	10pJ/bit/m^2
e_{amp}	0.0013pJ/bit/m^4
d_0	87m
E_{DA}	5*10-9nJ/bit
Initial energy	0.05J
Packet size / bits	500bits

B. PERFORMANCE COMPARISON AND ANALYSIS

1) DB INDEX EVALUATION

This paper introduces the DB index [31] evaluation value to judge and measure the effectiveness of the clustering algorithm. Intra-class dispersion and inter-class clustering are often used to judge the effectiveness of clustering. The DB index criterion simultaneously employs the inter-class clustering and the intra-class dispersion. Through calculating the index, we could determine which clustering method is the most reasonable.

① Intra-class dispersion: $S_i = \frac{1}{|C_i|} \sum_{X \in C_i} \|X - Z_i\|$, in which Z_i is the class center of class C_i and $|C_i|$ represents the number of samples in class C_i.

② Inter-class distance: $d_{ij} = \|Z_i - Z_j\|$, which employs the distance between the two class center to represent the inter-class distance.

③ DB index:

$$DB_k = \frac{1}{k} \sum_{i=1}^{k} R_i \tag{31}$$

where k is the cluster amount. According to DB index evaluation, the lower the DB_k value is, the better performance the clustering plays. R_i could be expressed as:

$$R_i = \max_{j=1,\ldots,k, j \neq i} \frac{S_i + S_j}{d_{ij}} \tag{32}$$

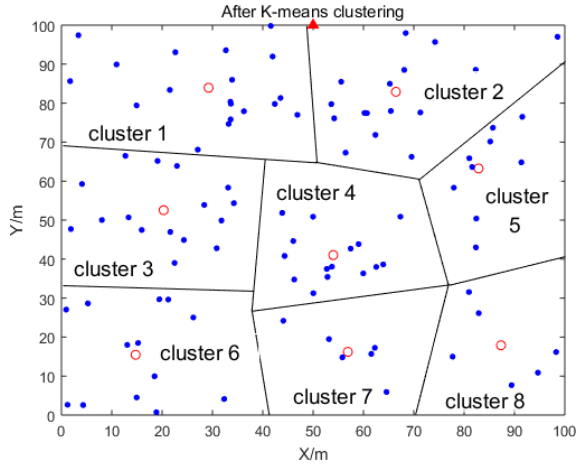

FIGURE 10. The condition before splitting and merging operations.

2) ALGORITHM PERFORMANCE ANALYSIS

a: SPLITTING AND MERGING OPERATION EVALUATION

There are 100 nodes deployed in a (100*100) m^2 network area. Figure 10 shows the network clustering before splitting and merging operations. As illustrated by the figure, the clustering structure cannot avoid the impact of ''energy hole'', i.e., in the condition where Cluster 1 area close to the base station (The solid triangle in red color) contains too many nodes and consumes too much energy, the data transmission of the network is affected correspondingly. Figure 11 shows the node cluster distribution after splitting and merging operations. Cluster 1 and Cluster 2 areas in Figure 10 are split. Cluster 7 and Cluster 8 areas in Figure 10 are merged. The radius of the cluster domain in the entire network meets the minimum energy consumption model under multi-hop transmission, which could effectively prolong the network life cycle.

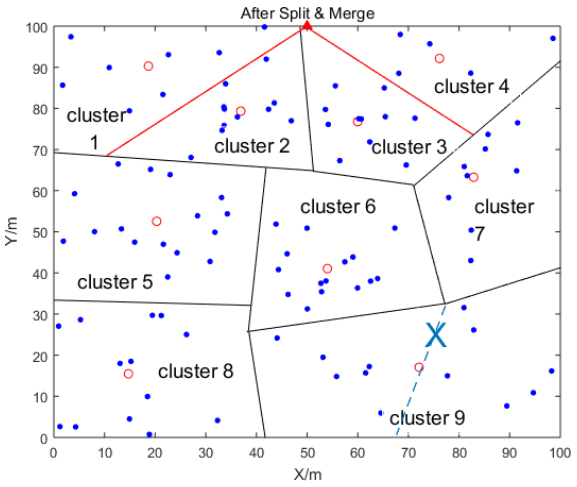

FIGURE 11. The condition after splitting and merging operations.

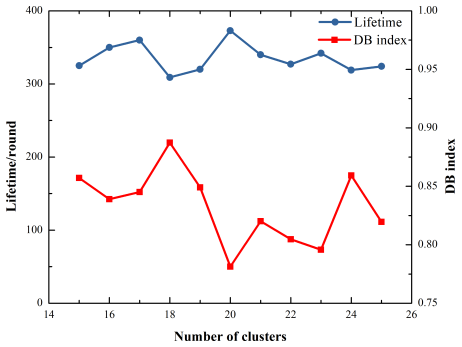

FIGURE 12. Number of clusters.

b: DETERMINED THE NUMBER OF CLUSTERS

Splitting and merging operations are mainly used to solve the energy hole problem. That is, in a multi-hop transmission route, the energy consumption of a cluster head node closer to a base station is higher than that of other nodes, which leads to a decrease in the life cycle of the nodes near the base station and then affecting the entire network service cycle. Therefore, during the splitting and merging operations, taking the splitting operation as the core act would make the k value increase accordingly. At this moment, given the values larger than the initial k value, it is feasible to control k value through the splitting and merging operation in the consideration of the influence of the cluster amount onto the DB index evaluation and network life cycle. The UCATD algorithm proposed in this paper derives the optimal k value as 18 with the lowest energy consumption Equation (13) in the clustering process. It can be seen in Figure 12 that when the number of clusters is 20, the DB evaluation value is at the minimum value and the network life is high. The network life cycle achieves the maximum value, so the value of the parameter c in Equation (20) can be changed to control the number of final clusters. After splitting and merging operations, the value of k turns to 20.

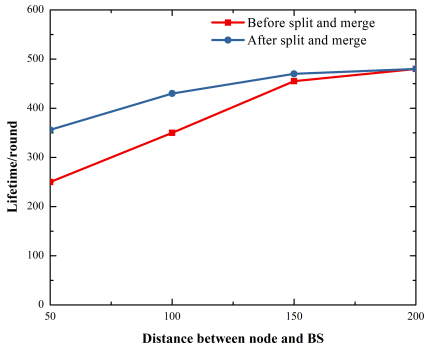

FIGURE 13. The influence of splitting and merging operations onto node life cycle.

c: INFLUENCE OF SPLITTING AND MERGING OPERATIONS ONTO NODE LIFE CYCLE

Figure 13 shows the comparison of the number of survival rounds of the nodes (distance from the base station) at different positions before and after splitting and merging operations. As illustrated in the figure, the lifetime of nodes from the BS 50m is increased by 42%, when it comes to 100m, the lifetime of nodes is increase by 23%. Thus, the splitting and merging operations significantly improves the survival time of nodes near the base station and eliminates the "energy hole" effect on the network lifetime in the multi-hop transmission model, and the operations effectively improves the network life cycle.

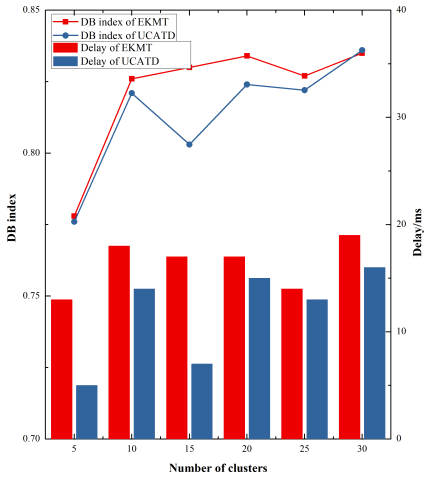

FIGURE 14. Clustering evaluation.

3) COMPARISON OF ALGORITHM PERFORMANCE

a: CLUSTERING EVALUATION

The UCATD algorithm proposed in this paper uses the maximum distance method when selecting initial k values in the clustering process. The EKMT algorithm adopts the random selection method in the original K-means algorithm when k initial values are selected. As can be seen from Figure 14, corresponding to the different cluster numbers k, the DB

evaluation value of the proposed UCATD algorithm is smaller than the DB evaluation value of the EKMT algorithm, which implies the clustering performance of UCATD is better. Meanwhile, the average delay during executing clustering by the UCATD algorithm is shorter than that by the EKMT algorithm. The UCATD algorithm thus could reduce the delay of the entire network and improve the data transmission efficiency.

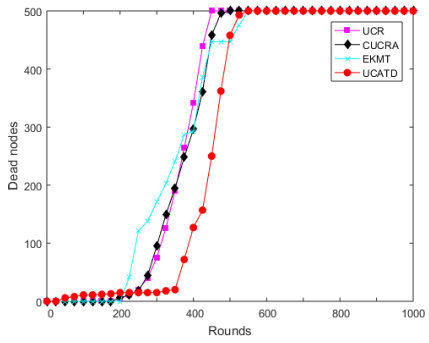

FIGURE 15. Network life cycle.

b: NETWORK LIFECYCLE

Figure 15 shows the network life cycle comparison curves for the four highlighted algorithms. It can be seen from the figure that the network life cycle of the UCATD algorithm is longer than that of other comparison algorithms. The network lifetime is prolonged to 19%, 21% and 65% when it compares with UCR, CUCRA and EKMT. Although the round amount of the last dead node of the EKMT algorithm is higher than that of the UCATD algorithm, the round amount interval from the first-appeared node death of the UCATD algorithm to the last-appeared node death is relatively low, which means that the energy consumption of all nodes in the network is balanced, and the effective life cycle of the entire network is extended by the UCATD algorithm.

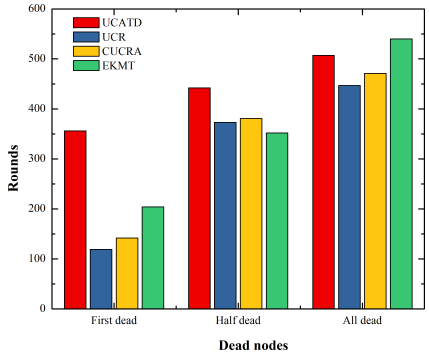

FIGURE 16. The comparison of the dead nodes.

c: COMPARISON OF DEAD NODES

Figure 16 shows the number of rounds for the first dead node appears, the number of rounds for the half number of dead nodes appear, and the number of rounds for all of nodes die. Comparison among the UCATD algorithm, the UCR algorithm, the CUCRA algorithm, and the EKMT algorithm in form of histogram, it can be seen from the figure that the UCATD algorithm has a longer network cycle than the other algorithms in the three cases, especially referring to the number of rounds when the first dead node appears, the result indicates that the UCATD algorithm could prolong the node survival time and effectively extend the network life cycle.

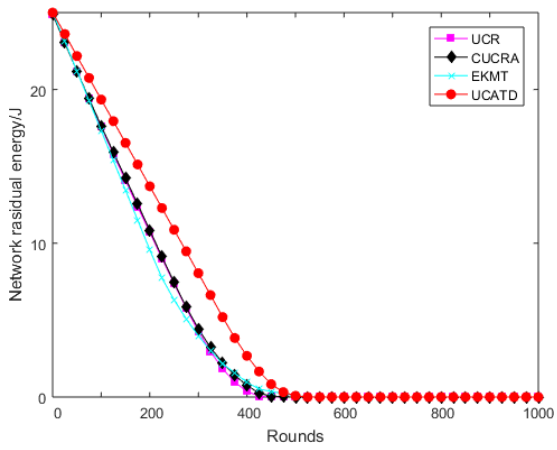

FIGURE 17. The comparison of network residual energy.

d: NETWORK RESIDUAL ENERGY

Figure 17 shows the network residual energy comparison curves for the four algorithms. It can be seen from the figure that the UCATD algorithm has the most remaining energy in each rounds, followed by UCR and CUCRA and the least is the EKMT algorithm. The network residual energy curve reflects the energy consumption of the four algorithms and is consistent with the network life cycle curve shown in Figure 15.

e: NETWORK THROUGHPUT

Figure 18 shows the comparison curves of the total data volume transmitted by the four algorithms during the network life cycle. It can be seen from the figure that the UCATD algorithm has the largest amount of data transmission and the EKMT algorithm has the least amount of data transmission, which verifies the validity of the data fusion tree constructed in this paper and the UCATD algorithm could improve the throughput of the network by assigning a reasonable timestamp to the cluster nodes. The throughput is increased by up to 20%, 19% and 68% when comparing with UCR, CUCRA and EKMT.

f: CLUSTER HEAD ELECTION

Figure 19 shows the comparison of the number of nodes' being elected as heads under the UCR algorithm, CUCRA algorithm, EKMT algorithm, and UCATD algorithm. It can be seen from the figure that the UCR and CUCRA algorithms

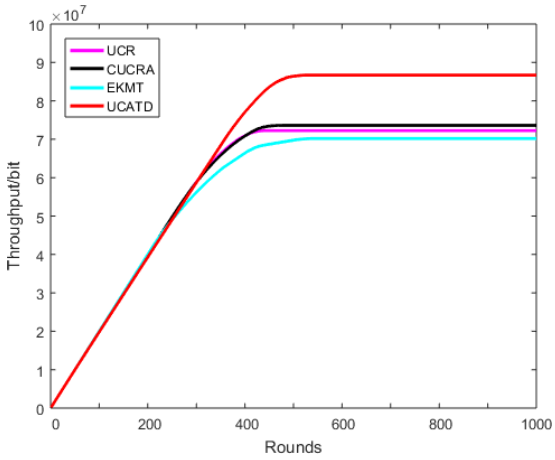

FIGURE 18. Network throughput.

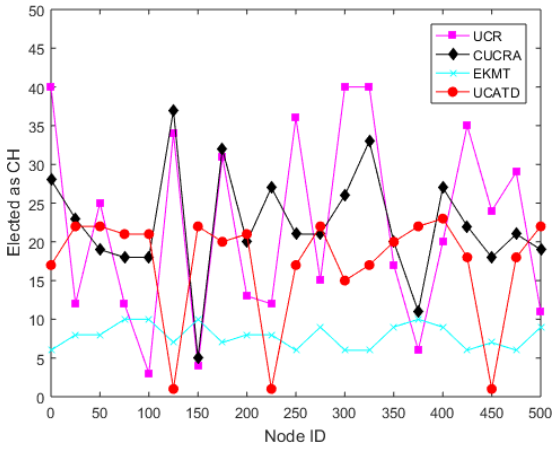

FIGURE 19. The number of nodes' being elected as head.

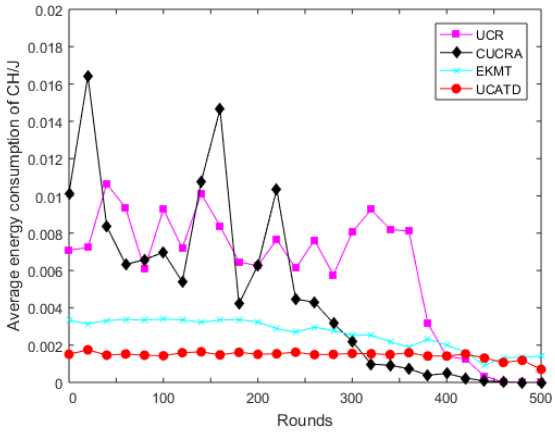

FIGURE 20. Average energy consumption of cluster head node.

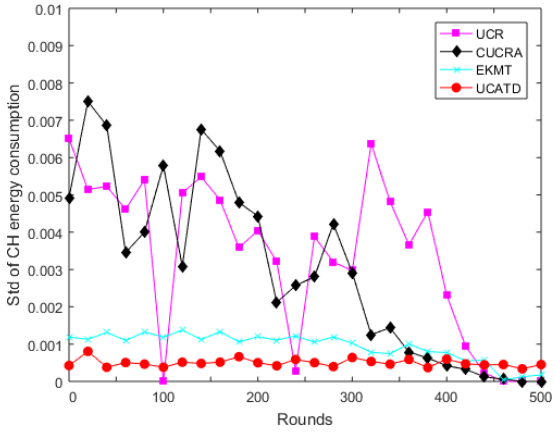

FIGURE 21. Energy consumption standard deviation of cluster head nodes.

have a wide range of variation. The EKMT algorithm and the UCATD algorithm have a smaller range of variation. This indicates that the UCR algorithm and the CUCRA algorithm lack the consideration of the remaining energy of the nodes during choosing cluster heads. If one node is frequently elected as cluster heads, its corresponding number of survival rounds will also decrease. The UCATD algorithm is based on k-means clustering, which adopts the intra-cluster head election, and simultaneously considers the location information of the nodes, the cluster center points and the base station. Meanwhile, the consideration also integrates the remaining energy factor to balance the energy consumption of each node in the network. In this way, the early death of some nodes due to the high number of being elected as cluster heads could be avoided.

g: AVERAGE ENERGY CONSUMPTION OF CLUSTER HEAD NODE

Figure 20 shows a comparison of the average energy consumption of cluster head nodes by the four algorithms. It can be seen from the figure that the cluster head energy consumption of the UCATD algorithm is lower than that of the UCR algorithm, CUCRA algorithm and EKMT algorithm, which verifies the effectiveness of cluster head distribution of the UCATD algorithm.

h: ENERGY CONSUMPTION STANDARD DEVIATION OF CLUSTER HEAD NODES

Figure 21 shows the standard deviation of cluster head node energy consumption by the four algorithms in different rounds. It can be seen from the figure that the UCATD algorithm makes the energy consumption of cluster heads more balanced, while the UCR algorithm and CUCRA algorithm select the cluster heads in a random manner, and the cluster head energy consumption standard deviation fluctuates greatly.

VI. CONCLUSIONS

As an underlying technology of IoT, the development of WSNs technology makes the application of IoT widespread. For solving the problem of unbalanced energy consumption and data transmission delay in WSNs, an unequal clustering concerned with time-delay routing protocol is proposed. This paper firstly proposes an improved K-means algorithm. Aiming at resolving the difficulty of determining the number of

cluster heads, a method for calculating the optimal number of cluster heads in multi-hop transmission mode is proposed. In the clustering stage, the initial center is selected by region division method to make the clustering more uniform which brings the improved efficiency of the algorithm. Concerning the problem of ''energy hole'' caused by non-uniform clustering, splitting and merging operations are introduced to make the energy consumption of nodes in the network more evenly. To decrease transmission delay in the process of data fusion, we construct a data fusion tree to maximize the time slots utilization. At the same time, the fusion tree construction algorithm is adopted independently. Simulation experiment results show that the proposed unequal clustering routing protocol improves the performance of the network and balances network energy consumption, and thus extends the network lifetime. The proposed algorithm is especially suitable for the delay-limited applications of IoT.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to the editors and the anonymous reviewers for their helpful comments.

REFERENCES

[1] J. Li, L. Huang, Y. Zhou, S. He, and Z. Ming, ''Computation partitioning for mobile cloud computing in a big data environment,'' *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2009–2018, Aug. 2017.

[2] W. Wei, X. Fan, H. Song, X. Fan, and J. Yang, ''Imperfect information dynamic Stackelberg game based resource allocation using hidden Markov for cloud computing,'' *IEEE Trans. Services Comput.*, vol. 11, no. 1, pp. 78–89, Jan./Feb. 2018.

[3] H. Chen, X. Xie, W. Shu, and N. Xiong, ''An efficient recommendation filter model on smart home big data analytics for enhanced living environments,'' *Sensors*, vol. 16, no. 10, p. 1706, 2016, doi: 10.3390/s16101706.

[4] J. S. Peng and Y. M. Shao, ''Intelligent method for identifying driving risk based on V2V multisource big data,'' *Complexity*, vol. 2018, May 2018, Art. no. 1801273.

[5] F. Han, S. Zhao, L. Zhang, and J. Wu, ''Survey of strategies for switching off base stations in heterogeneous networks for greener 5G systems,'' *IEEE Access*, vol. 4, pp. 4959–4973, 2016.

[6] J.-Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, ''Industrial Internet: A survey on the enabling technologies, applications, and challenges,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1504–1526, 3rd Quart., 2017.

[7] M. K. Yapici and T. E. Alkhidir, ''Intelligent medical garments with graphene-functionalized smart-cloth ECG sensors,'' *Sensors*, vol. 17, no. 4, p. 875, 2017, doi: 10.3390/s17040875.

[8] W. Wei, H. Song, W. Li, P. Shen, and A. Vasilakos, ''Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network,'' *Inf. Sci.*, vol. 408, pp. 100–114, Oct. 2017.

[9] J.-Q. Li, S.-P. Zhang, L. Yang, X.-H. Fu, Z. Ming, and G. Feng, ''Accurate RFID localization algorithm with particle swarm optimization based on reference tags,'' *J. Intell. Fuzzy Syst.*, vol. 31, no. 5, pp. 2697–2706, 2016.

[10] J.-Q. Li, S.-Q. He, Z. Ming, and S. Cai, ''An intelligent wireless sensor networks system with multiple servers communication,'' *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, p. 960173, 2015.

[11] O. O. Ogundile and A. S. Alfa, ''A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks,'' *Sensors*, vol. 17, no. 5, p. 1084, 2017, doi: 10.3390/s17051084.

[12] Z. Xu, L. Chen, T. Liu, L. Cao, and C. Chen, ''Balancing energy consumption with hybrid clustering and routing strategy in wireless sensor networks,'' *Sensors*, vol. 15, no. 10, pp. 26583–26605, 2015.

[13] M. Bagaa, M. Younis, D. Djenouri, A. Derhab, and N. Badache, ''Distributed low-latency data aggregation scheduling in wireless sensor networks,'' *ACM Trans. Sensor Netw.*, vol. 11, no. 3, p. 49, 2015, doi: 10.1145/2744198.

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, ''Energy-efficient communication protocol for wireless microsensor networks,'' in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, pp. 1–10.

[15] B. S. Lee, H.-W. Lin, and W. Tarng, ''A cluster allocation and routing algorithm based on node density for extending the lifetime of wireless sensor networks,'' *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 51–62, 2012, doi: 10.1109/WAINA.2012.42.

[16] Z. Xu, Y. Yin, and J. Wang, ''An density-based energy-efficient routing algorithm in wireless sensor networks using game theory,'' *Int. J. Future Gener. Commun. Netw.*, vol. 5, no. 4, pp. 60–70, 2012.

[17] C. Li, M. Ye, G. Chen, and J. Wu, ''An energy-efficient unequal clustering mechanism for wireless sensor networks,'' in *Proc. IEEE Int. Conf. Mobile Ad Hoc Sensor Syst. Conf.*, Nov. 2005, pp. 604–611.

[18] 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, 2009.

[19] W. Tong, W. Jiyi, X. He, Z. Jinghua, and C. Munyabugingo, ''A cross unequal clustering routing algorithm for sensor network,'' *Meas. Sci. Rev.*, vol. 13, no. 4, pp. 200–205, 2013.

[20] R. Elkamel and A. Cherif, ''Energy-efficient routing protocol to improve energy consumption in wireless sensors networks,'' *Int. J. Commun. Syst.*, vol. 30, no. 17, p. e3360, 2017.

[21] L. Tan, Y. Gong, and G. Chen, ''A balanced parallel clustering protocol for wireless sensor networks using K-means techniques,'' in *Proc. Int. Conf. Sensor Technol. Appl.*, Aug. 2008, pp. 300–305.

[22] B. Jain, G. Brar, and J. Malhotra, ''EKMT-k-means clustering algorithmic solution for low energy consumption for wireless sensor networks based on minimum mean distance from base station,'' in *Networking Communication and Data Knowledge Engineering*. Singapore: Springer, 2018, pp. 113–123.

[23] T. Li, R. Feng, Z. Fan, J. Wang, and J.-U. Kim, ''An improved PEGASIS protocol for wireless sensor network,'' in *Proc. Int. Conf. Comput. Comput. Sci.*, Oct. 2016, pp. 16–19, doi: 10.1109/COMCOMS.2015.20.

[24] S. C. Huang, P. J. Wan, C. T. Vu, Y. Li, and F. Yao, ''Nearly constant approximation for data aggregation scheduling in wireless sensor networks,'' in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2007, pp. 366–372.

[25] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang, ''A delay-efficient algorithm for data aggregation in multihop wireless sensor networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 1, pp. 163–175, Jan. 2011, doi: 10.1109/TPDS.2010.80.

[26] L. Ma, J. Liu, and J. Luo, ''Method of wireless sensor network data fusion,'' *Int. J. Online Eng.*, vol. 13, no. 9, pp. 114–122, 2017, doi: 10.3991/ijoe.v13i09.7589.

[27] M. Bagaa, A. Derhab, N. Lasla, and A. Ouadjaout, ''Semi-structured and unstructured data aggregation scheduling in wireless sensor networks,'' in *Proc. INFOCOM*, Mar. 2012, pp. 2671–2675, doi: 10.1109/INFCOM.2012.6195676.

[28] C.-T. Cheng, K. T. Chi, and F. C. M. Lau, ''A delay-aware data collection network structure for wireless sensor networks,'' *IEEE Sensors J.*, vol. 11, no. 3, pp. 699–710, Mar. 2011, doi: 10.1109/JSEN.2010.2063020.

[29] M. Koupaee, M. R. Kangavari, and M. J. Amiri, ''Scalable structure-free data fusion on wireless sensor networks,'' *J. Supercomput.*, vol. 73, no. 12, pp. 5105–5124, 2017, doi: 10.1007/s11227-017-2072-0.

[30] C. M. Chao and T. Y. Hsiao, ''Design of structure-free and energy-balanced data aggregation in wireless sensor networks,'' *J. Netw. Comput. Appl.*, vol. 37, no. 1, pp. 229–239, 2014.

[31] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, ''Understanding of internal clustering validation measures,'' in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 911–916.

XIN FENG received the M.Sc. degree from the School of Computer Science and Information Technology, Northeast Normal University, China, in 2006. He is currently an Associate Professor with the College of Computer Science and Technology, Changchun University of Science and Technology. His research area includes Internet of Things, network security, and body area networks.

JING ZHANG received the M.Sc. and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 2012 and 2015, respectively. She is currently a Lecturer with the College of Computer Science and Technology, Changchun University of Science and Technology. Her research interests include wireless sensor networks and complex networks.

CHENGHAO REN is currently pursuing the master's degree with the College of Computer Science and Technology, Changchun University of Science and Technology. His research interests are wireless sensor network and clustering algorithm.

TINGTING GUAN is currently pursuing the master's degree with the College of Computer Science and Technology, Changchun University of Science and Technology. Her research interests are wireless sensor network and data fusion.