Research Article

A High Efficient and Real Time Data Aggregation Scheme for WSNs

Tao Du, Zhe Qu, Qingbei Guo, and Shouning Qu

1 School of Information Science and Engineering, University of Jinan, Jinan 250022, China
2 Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250014, China
3 Shandong Provincial Key Laboratory of Computer Networks, Jinan 250014, China

Correspondence should be addressed to Zhe Qu; alicequz@hotmail.com

Academic Editor: Ying-Hong Wang

Copyright © 2015 Tao Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data aggregation is an important method of improving transmitting efficiency of WSNs, but existing researches have some disadvantages as follows: several periods delay will be generated when filtering messages; the reduplicated messages filtering ratio is low; complex calculations need to be executed; and extra hardware should be added to obtain high performance. To resolve these problems, this paper proposes a real time and high efficient data aggregation scheme (dynamical message list based data aggregation, DMLDA) based on clustering routing algorithm. DMLDA includes three procedures: activating nodes, clustering nodes, and filtering messages. In filtering procedure, a special data structure named dynamic list will be established in every filtering node; it is designed to store messages transmitted by filtering node, and, comparing current message with list's items, the message's redundancy can be judged without any delay. To improve the filtering efficiency, the message list can be adjusted dynamically. The three procedures of data aggregation are all introduced, and the filtering method is designed in detail in this paper. At last, a series of experiments are simulated to prove our scheme's performance, and the advantages are analyzed in theory.

1. Introduction

WSNs (wireless sensor networks) are a new technology used to collect information from harsh environment; it is developing very rapidly and now it is one of the most important fields of information science. In WSNs, there are one or several sink nodes and a lot of sensor nodes. Sink node has limitless energy and powerful calculation capability, while sensor node has limited energy and poor calculation capability [1–5]. For the hardware restrict of sensor node, how to use limited resource to transmit more useful messages is the most important specification [6]. But, as the distributed and self-organized structure of WSNs, the transmitting efficiency is much lower than classical networks. Therefore many methods have been proposed to improve the transmitting efficiency of WSNs. Among these methods, data aggregation is one of the most important ones by reducing invalid message transmitting. There exist many data aggregation algorithms, but these algorithms have many disadvantages: real time is poor; the efficiency of filtering is not high; the burden of extra hardware is too large, and so forth. In this paper, a new high efficient and real time data aggregation scheme is proposed to resolve existing problems, and this scheme includes three main steps: activating all sensor nodes by ladder diffusion, which is proposed in our previous research in [7]; clustering all nodes based on semistatic HAC method, which is proposed in our previous research in [8]; and filtering reduplicated messages using dynamical list, which is the key of this paper. The rest of this paper is organized as follows: in Section 2, we analyze the performance standards and introduce some of the main existing research works; in Section 3, we introduce the design of three steps of DMLDA in detail; in Section 4, we prove and analyze the performance of DMLDA in theory and by simulated experiments; in Section 5, the innovations and advantages of this paper are concluded.

2. Related Works

2.1. The Key Problems of Designing Data Aggregation Schemes for WSNs. Nowadays, WSNs have been widely used in
battlefield surveillance, industrial production detection, and environmental monitoring, and, in these applications, information has strong timeliness, so the transmission delay should be as little as possible. In WSNs, delay occurs mainly in data aggregation when messages are waiting to be judged their repetitiveness and whether to be filtered, so how to reduce delay in data aggregation is one of the most important problems in designing scheme, and real time filtering procedure is the key performance of data aggregation.

Meanwhile, according to the structure of WSNs, when designing data aggregation scheme, the extra burden both in hardware and in software should also be taken into account for sensor nodes' limited memory and poor capability of computing. Therefore the scheme should be operated by as less calculation as possible and should not need much storage space without causing degradation in performance of filtering.

In data aggregation, the distribution of filtering nodes is also an important problem, and it will very much affect the performance of data aggregation. In WSNs, the filtering nodes must be held by sensor nodes; however all sensor nodes are randomly deployed in sensing area and each sensor node cannot obtain global information of WSNs. Therefore, how to optimally distribute filtering nodes without global information is a basic problem to be resolved in designing data aggregation scheme.

Besides, on the demands of real time transmission, little extra burden, and distributed structure, some quantitative standards should be considered when designing data aggregation scheme. The key function of data aggregation is to filter reduplicated messages to reduce the invalid transmitting, so one of the most important standards of judging the performance of data aggregation is the efficiency of filtering, which includes two parameters as the accuracy of filtering and omission rate of reduplicated messages:

\[
\text{Accu} = \frac{n_r}{N_s},
\]

The accuracy of filtering is defined in (1), as the ratio of the number of real reduplicated messages \(n_r\) should be filtered and the number of all massages \(N_s\) filtered. This parameter's value is higher; more right messages will not be filtered, and WSNs' normal transmitting function will be not much disturbed by the filtering procedure:

\[
\text{Omi} = \frac{n_r}{N_o}.
\]

The omission rate is defined in (2), as the ratio of the number of messages which should be filtered but is wrongly transmitted \(n_r\) and the number of all reduplicated messages \(N_o\). This parameter's value is higher; the number of invalid transmitting will be less, and WSNs' energy will be much conserved.

2.2. Existing Researches. For the broad application prospect of WSNs, many researchers have focused on data aggregation, and a lot of schemes have been proposed. These researches can be divided into two types: one is operated in data link layer and based on MAC protocol; the other is operated in network layer and based on routing protocol. The schemes based on routing protocol have these advantages: it can be dynamically adjusted according to routing states; it is easy to be deployed than the one based on MAC protocol [6]; it is fit for more application situations and has less restriction for WSNs structure. Most of new proposed researches are based on routing protocol, and these schemes will be, respectively, introduced according to their routing methods and policy of filtering messages in this section.

The efficiency of data aggregation schemes operated in network layer is much influenced by their routing method. The routing method can be divided into three types: data-centric, tree-based, and clustering-based. In data-centric routing, messages are transmitted by flooding mode, and the reduplicated messages are filtered by some special nodes that will be chosen to act as filtering by sink node according to WSNs global distribution [9, 10]. Although this scheme is simple to be designed and has high self-organization ability and robustness, it has two serious defects: too many reduplicated messages will be generated in WSNs which will influence the efficiency of routing; and the distribution of filtering nodes violates the principle of distributed characteristic of WSNs. In tree-based routing, a spanning tree will be generated from source nodes to sink node, and the message will be transmitted toward this tree from leaf nodes to root node. In the spanning tree, some special nodes will be chosen as filtering nodes. The key of this scheme is to efficiently establish the spanning tree [11, 12]. Typical researches like Steiner minimum tree (SMT) that sensor nodes' residual energy will be considered and the nodes with more energy will have larger probability to be chosen as filtering nodes and in PEDAP that the filtering nodes are all set on parent nodes of spanning tree [13], which is easy to be applied but needs more filtering nodes than necessary. In general, the filtering efficiency of tree-based routing can be improved by optimizing the distribution of filtering nodes, but this type of method has the defects those the procedure of routing is not flexible for the single transmitting direct through the spanning tree, and the calculation of establishing spanning tree is too complicated for WSNs.

Clustering-based routing is one of the most popular methods in WSNs. In this routing, all sensor nodes will be divided to a certain number of clusters by special rule, and in every cluster one node will be elected to act as cluster head. The cluster head is responsible for collecting messages from other nodes in its cluster and forwarding these messages to sink node. After several rounds transmitting, cluster head will be rotated to other nodes. The advantages of this kind of routing are that the task distribution among different sensor nodes is more balanced, and the method has strong robustness. But this routing is relatively complex in that it is hard to design, and its efficiency is influenced by clustering method. Now there are some researches focusing on combining filtering function with clustering routing procedure such as LEACH [14], LEACH-C [15], and HEED [16]. In these schemes, cluster heads are always chosen to act as filtering nodes, so how to select cluster head and rotate it is the key factor of affecting filtering efficiency. But, in existing schemes,
the picking and rotation of cluster head are all designed based on simple probability, which leads that the distribution of filtering nodes is not optimal and cannot be adjusted according to transmission of WSNs.

Besides the restrictions mentioned above, existing schemes have defect in the work mode of filtering messages. These schemes adopt the method of transmission delaying: when filtering nodes receive a new message, it will delay several periods to find whether subsequent messages are reduplicated with current one and then to judge to transmit this message or abandon it. To improve the filtering ratio and reduce the hardware burden, the delaying policies are developed from simple static delaying periods to dynamical delaying periods which are adjusted according to WSNs transmission [17–19]. The advantages of transmission delaying include that it takes a little hardware burden; the calculation of filtering is simple; and the filtering ratio is relatively high based on long delay. But this method has two defects that the real time performance of data aggregation is decreasing and that the dynamical adjustment of delaying period is hardly obtained optimal result for the distributed structure of WSNs.

Based on analysis of existing researches, although many data aggregation schemes have been proposed, the demands of real time transmission, little extra burden, distributed structure, and high filtering efficiency are not well satisfied. How to improve real time, reduce the computational complexity of distributed structure, and make it more efficient is the emphasis of designing data aggregation scheme.

3. Detail of Design

According to the former analysis, we will propose a new clustering-based and real time data aggregation scheme named DMLDA in detail in this section. In this scheme, a special data structure named dynamical list is deployed in every filtering node to store history messages before transmission, and we use the message list replacing the periods delaying method to filter reduplicated messages which highly improves real time performance. The length of this list will be initially set according to the hops from filtering node to sink node, and, with transmission running, it will be adjusted by filtering state. In this scheme, there are three parts functions being implemented: first is to activate all sensor nodes; second is to cluster all sensor nodes including of choosing cluster heads and rotating them; third is to design the method of filtering reduplicated messages. To achieve these functions, the process of DMLDA is divided to three steps: in the first step a ladder diffusion method is designed to activate all nodes and at the same time the adjacent relationship of every node will be established; in the second step a semistatic hierarchical routing algorithm based on HAC is designed to organize the routing of WSNs, and the energy consumption in all nodes can be balanced by this routing algorithm; in the third step filtering node will be deployed in every cluster head, and the content and length of message list in filtering node will be dynamically adjusted. We will introduce the detail design in the following section.

### Table 1: The definition of packet head.

| Field  | Description                           |
|--------|---------------------------------------|
| ID     | ID of the node which sends this packet |
| Pkt-typ | Packet type (LAD/CON/DATA)            |
| Gra    | Means the hops number to sink node     |
| Ene    | The node's residual energy             |
| Axis   | The axis of node ID                    |

### Table 2: The structure of neighbor node list.

| Field  | Description                           |
|--------|---------------------------------------|
| ID     | ID of current node in LNN              |
| L-Next | The next node in LNN                   |
| Ene    | The node's residual energy             |
| Axis   | The node's axis                        |

3.1. The Activation of All Sensor Nodes. After WSNs deployment, all sensor nodes should be activated to obtain the location of sink node and discover its neighbor nodes, and, in DMLDA, this step should also achieve the function of computing the hops number from every node to sink node to help establish message list. To obtain these targets, a special packet head is designed as Table 1, and the activating packet will be transmitted by the method of ladder direct diffusion. At the same time, a list storing all neighbor nodes will be created in every node when it receives activating packet, and the list's structure is designed as Table 2. The content activating packet’s head and neighbor nodes’ list will be updated when transmission is running.

The first step is that the sink node creates and broadcasts some activated packets and sets these packets’ grade value to 1. If a node receives the packet, it will set its grade value to 1 and add sink node to its neighbor node list. Then the current node will update the activated packet with its location and energy information, set the grade value of itself as 1, and update the packet’s value as 2. The procedure is shown as Figure 1.

In the second step, activated nodes will broadcast the new packet along the reverse direct of source node. When a node receives an activating packet, it will extract the grade value from packet head and compare the value with itself. If the packet value is larger, it will be abandoned; otherwise, the node's grade value will be updated by packet value. While comparing the grade value, the neighbor nodes list will also be updated: the list will be browsed to judge the source node whether it is in the list. If the node is not in list, the source node will be added to list and create a confirming packet with current node's information and send it to source node; otherwise, no operation will be executed.

When a node is receiving a confirming packet, it will browse its neighbor node list and judge how to deal with this packet. If the source node has been in list, no operation will be executed and the packet will be abandoned; otherwise the source node will be added to the list and create a confirming packet with current node's information and return it to source node.

While updating the nodes’ information, a new activating packet will be generated with current node's information. And then this packet will be sent to next hop nodes. This
operation will be executed until all nodes in WSNs have been activated, and the hops to sink node are all stored in each node, and all neighbor nodes have been found too. The procedure of this step is shown in Figures 2 and 3.

### 3.2. The Clustering of All Sensor Nodes

Based on the result of step 1, all sensor nodes are clustered by a semistatic HAC method according to their location and residual energy. Besides basic efficient routing function, there are four main targets that should be achieved in this step: nodes should be efficiently clustered; the cluster head should be reasonably elected; all cluster heads should be dynamically rotated; and the distribution of clusters should be adjusted according to nodes' state. In this step, a special data structure named LNC (list of nodes in cluster) has been defined in Table 3, and LNC is created and maintained by cluster head and it stores the information of member nodes.

In this clustering step, the distance of each pair of nodes can be calculated by (3) and the SEPC (the summary of every pair of nodes in cluster) of node $i$ is calculated by (4). And, in HAC, smaller clusters will be merged to be a larger one, and the standard of merging is the distance of each pair of clusters which could be calculated by

\[ D_{ab} = \frac{2E}{E_a + E_b} \sqrt{(X_a - X_b)^2 + (Y_a - Y_b)^2}, \quad (3) \]

\[ C(i) = \sum_{j=1, j\neq i}^{m} D(i, j), \quad (4) \]

\[ D_{PGAB} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} D(i, j). \quad (5) \]

In (3), $E$ is the initial energy of sensor nodes and $E_a$ and $E_b$, respectively, mean the residual energy of node $a$ and node $b$. $X_a$ and $Y_a$ represent the coordinate of node $a$, and $X_b$ and $Y_b$ represent the coordinate of node $b$. In (4), $m$ is the number of nodes in current cluster. In (5), $m$ and $n$, respectively, mean the number of nodes in cluster $A$ and cluster $B$, and $D(i, j)$ is
the distance between node $i$ in cluster $A$ and node $j$ in cluster $B$, and the value of this formula means the average distance from all nodes in cluster $A$ to all nodes in cluster $B$.

Based on these formulas and data structures, all sensor nodes can be clustered according to nodes’ residual energy, and cluster head will be dynamically rotated. The detail of clustering and cluster head rotation procedure is proposed in our previous researches, and the illustration of clustering is shown in Figures 4–6, in which nodes’ residual energy has little difference.

To better illustrate the procedure of clustering-based routing algorithm, a simple WSNs model is defined with 9 sensor nodes randomly distributed in sensing area. In the first step of clustering, each node forms a cluster, acting as the cluster head. A LNC will be created in every node. The result is shown in Figure 4.

In Figure 5, the clusters with least distance will be merged according to the distance calculated by (5), and LNC of each smaller cluster will be combined, and the nodes’ order in LNC will be sequenced by the value of SEPC which is computed by (4). Repeat this operation until all distance of clusters has been larger than the threshold defined according to WSNs scale.

In Figure 6, the first round of clustering procedure is finished and WSNs will come into sensing and transmitting phase. After several rounds transmitting, the energy consumption of cluster head will be much larger than their member nodes, and then the rotating procedure will execute with the order in LNC. After some rounds of rotating, the difference of nodes’ residual energy is larger and just rotating cluster head cannot balance this difference, so a reclustering operation will be executed, and the result is shown in Figure 7. Comparing the result of Figure 6 with Figure 7, it can be concluded that the clusters’ distribution can be dynamically adjusted according to nodes’ residual energy, so the routing algorithm can improve the energy efficiency of WSNs.

- **Figure 4**: The first step of clustering.
- **Figure 5**: The result of one round of merging.
- **Figure 6**: The final result of clustering.
- **Figure 7**: The result of clustering with different nodes’ energy.

### Table 4: The structure of list.

| ID | ID of cluster head node |
|----|-------------------------|
| Head | The head of whole list |
| Length | The length of list |
| Flag | The threshold of changing length |
| Lifetime | The content’s time without updating |
| L-next | The next member of this list |
| Content | The information content of this member |

3.3. **The Design of Filtering Method.** After last two steps, all nodes have been activated with the information of location and hops from itself to sink node, and clustering-based routing structure has been established. In this step, filtering method will be designed in detail. To improve real time performance of filtering, a message list is designed as Table 4
which is created and maintained by each filtering node, and
the content and length of this list will be dynamically updated
according to transmitting state of filtering node.

The first operation in this step is to choose filtering
node and create message list in every filtering node. In
clustering-based routing, cluster heads will collect messages
from their member nodes and sometimes they will also
relay other clusters' messages, so cluster heads have obvious
advantage to act as filtering nodes. Because cluster heads
are randomly distributed in sensing area, and transmitting
burden of different cluster heads will be much different,
the nodes nearer to sink node should not only collect the
messages from their member, but also undertake more tasks
of relaying messages from outer nodes. In DMLDA, the hops
number from each node to sink node has been obtained in
activating phase that we set as the parameter grade value,
and the length of message list could be set according to this
value: the list length is shorter when hops number is larger,
which means the nodes faraway from sink node should store
less messages than nodes near sink node. The process of
this operation is as follows: firstly, the largest grade value of
all nodes $H_{level}$ will be discovered by browsing all clusters;
secondly cluster swill be browsed again to compute the
message list's length of each filtering node; and then the list
will be initialized. The pseudocodes of this process are as
shown in Procedure 1.

In this procedure, $H_{level}$ is the largest number of hops
from all sensor nodes to sink node, and $CH_{i,level}$ is
the hops number of cluster head node with ID $i$. In every list,
flag is a variable which means the frequency of message
in current item of List $i$, and lifetime is a variable means
that the time interval from now to last time that received


code

Procedure 1: Procedure filter nodes' configuration and list's initialization.

1. $H_{level} = 0$; //the current maximum level is set as 0.
2. Browse All Clusters to;
3. $CH_{i,level} = \text{get level of } CH_{i}$;
4. If $CH_{i,level} > H_{level}$:
5. $H_{level} = CH_{i,level}$;
6. end browse
7. Browse All CHs again;
8. $N_{level} = H_{level} - CH_{i,level}$;
9. //calculation of current head node's level
10. end browse;
11. //all head nodes' level has been initialed.
12. Browse All Clusters;
13. //Begin to establish the dynamic list
14. new List $i$,
15. List$ i$.length = $N_{level}$ + 1;
16. //the list's initial length is set according to its current level
for List$ i$.length
17. List$ i$.content = null;
18. //set the current list item as null;
19. List$ i$.next = new list;
20. List$ i$.flag = length;
21. List$ i$.lifetime = 0;
22. //Set the current item's threshold of updated as 0;
23. List$ i$ = List$ i$.next;
//Establish the list's link;
24. end for
25. end browse
//All filtering nodes' lists have been initialized;
26. if CH is being rotated
27. new list $j$;
//The current cluster's head has been changed to node $j$;
28. List$ j$ = List$ i$;
29. release List$ i$;
30. end if
31. if Clusters are being re-calculated
32. Filter nodes' configuration
33. Setup List;
34. end if
//When all nodes have been re-clustered, the List will be re-initialized;
35. End procedure


(1) a filter node ID, receives an information \( M_d \);
(2) browse List;
(3) List_\(i\) = List_\(i\).head;
(4) if List_\(i\).content == \( M_d \);
(5) Throw up this message
(6) List_\(i\).flag = List_\(i\).flag + 1;
//Prolong this message’s storing time;
(7) end if
(8) List_\(i\) = List_\(i\).next;
(9) if not exist any List_\(i\).content == \( M_d \);
(10) For all item of List_\(i\)
(11) flag = flag – lifetime;
//Reduce the threshold to help update list;
(12) all content lifetime + 1;
(13) change info_\(s\) to info_\(e\)
//Relay this message
(14) browse List;
(15) find the info with minimum flag
(16) List_\(i\).content = \( M_d \)
//Update list corresponding item
(17) List_\(i\).flag = List_\(i\).length
(18) List_\(i\).lifetime = 1;
(19) end if in line (9)
(20) end procedure

Procedure 2: Procedure data aggregation.

same message, and these two variables are used to control the dynamical adjustment of current list. When cluster head has been rotated, the list will be also transmitted to new cluster head. Through this procedure, the filtering node will be reasonably initialized according to its location.

The second operation is filtering and transmitting message. When all filtering nodes have been set and all message lists have been initialized, WSNs will come into the sensing and transmitting phase. When a filtering node receives a message \( M_d \), it will browse its message list List_\(i\) to judge \( M_d \) reduplicated or not: if there are some messages in List_\(i\) equaling \( M_d \), \( M_d \) will be thrown up and the content of List_\(i\) will be updated; if there are not any messages in List, equaling the \( M_d \), \( M_d \) will be added into List, and the whole message list will be updated. The pseudocodes of this process are as shown in Procedure 2.

In this procedure, if current item of List_\(i\) is equal to \( M_d \), the frequency of this message will add one, and it means the probability of deletion this item from List, will decrease. If there does not exist an item in List_\(i\) equaling \( M_d \), all item’s flag will minus its lifetime, and then the minimum flag in List, will be found, and this item will be replaced by \( M_d \), and new item’s flag and lifetime will be set as initial value. Through this procedure, content of message list can be dynamically updated, which is the basic function of filtering.

The third operation in this step is to dynamically update the length of message list. To achieve this function, two thresholds are defined: one is judging whether to reduce the length of list, and the other is judging whether to prolong the length of list, and the detail operation is shown in Procedure 3.

Table 5: Summary of parameters.

| Parameter                      | Value                  |
|-------------------------------|------------------------|
| Threshold distance \(d_0\) (m) | 10                     |
| \(E_{\text{elec}}\) (nJ/bit)  | 50                     |
| \(\epsilon_{s}\) (pJ/bit/m^2) | 10                     |
| \(\epsilon_{\text{mp}}\) (pJ/bit/m^4) | 0.0013                |
| Sensor node initial energy (J) | 1                      |

In this procedure, the List_\(i\) will be browsed and each item’s flag and lifetime will be compared with the threshold of reducing list’s length: if its lifetime is larger than list’s length and meanwhile its flag’s value has decreased to negative length, the item should be deleted in list without any replacement. If lifetime of List_\(i\) is no more than 1, and meanwhile the value of flag is greater than or equal to the length, a null item will be added to the end of List, Through this operation, the message list can be adjusted according to WSNs state, which will ensure that the hardware burden can be reduced while the filtering efficiency remains unchanged.

4. The Result of Experiments

In order to prove the performance of DMLDA, a series of experiments are designed and analyzed in this section, and these experiments are simulated by NS2. Some parameters of WSNs are listed in Table 5, and the sensing area is 1000*1000 m^2. Messages are generated in the sensor nodes, which are distributed at random locations in the sensing area. In our experiments, we compare the proposed algorithm, that is, DMLDA, with a typical data scheme using delay
international journal of distributed sensor networks

(1) a filter node $ID_i$ receives some information;
(2) browse List $i$
(3) for every item in list
(4) if List$_{i, next}.lifetime \geq \text{length}$ and List$_{i, next}.flag \leq 0$
  //Means this item should be deleted;
(5) length = length $- 1$;
(6) List$_{i, next} = \text{List}_{i, next}.next$;
(7) else if all List$_{i, next}.lifetime \leq 1$ and List$_{i, next}.flag \geq \text{length}$
  //Means all items’ content was often updated;
(8) length = length $+ 1$;
(9) new list
(10) List$_{i, next} = \text{new list}$;
(11) List$_{i}.flag = \text{length}$;
(12) List$_{i}.lifetime = 0$;
(13) List$_{i, next} = \text{List}_{i}$;
(14) end browse
(15) end procedure

Procedure 3: Procedure changing length of list.

transmitting mode based on LEACH routing. Moreover, theoretical analysis of the performance is provided.

4.1. Simulations. The standards of measuring data aggregation have been introduced in Section 2.1. Every new message should be compared with history messages in both of DMLDA and LEACH, so a new message cannot be mistaken as a reduplicated one. The experiments in this section are mainly used to measure the filtering ratio, which is defined as the percentage of filtered messages in all reduplicated ones by considering the additional burden and real time performance decline for WSNs. The additional burden it takes to WSNs can be compared according to the length of the list in DMLDA, and real time performance decline for WSNs can be quantified by computing the delaying cycle in LEACH. In order to demonstrate the performance of DMLDA comprehensively, three simulations are designed in our experiment: first, the change trend of filtering ratio with the increase of the proportion of the reduplicated messages is presented in Figure 8; then, Figure 9 shows the change trend of filtering ratio with the increase of sensor nodes; finally, we present the change of list’s length with the increase of reduplicated messages’ proportion, and the result is shown in Figure 10.

200 sensor nodes are deployed in the target area, and the change of filtering ratio is illustrated by three curves in Figure 8. The red and blue curves present the changing state with the message lists’ longest length is 4 and 7, respectively. The green curve shows the changing state of LEACH with the delaying cycle being 7 in every filtering node; that is, every message would be delayed 7 periods before being transmitted by the filtering nodes. The curve with the longest length of the message is 4 and shows that the filtering ratio increases when the reduplicated messages’ proportion increases from 10% to 30%, and it is lower than 90%; when the reduplicated messages’ proportion is larger than 50%, the filtering ratio is changed a little and the values are stabilized around 92%. When the length of the list is 7, it can be seen that the filtering ratio is almost stable and the lowest value is 93.7% when the proportion is 30%, and the highest value is 95.2% when the proportion is 50% and 60%. The result of LEACH shows that, with the proportion of the reduplicated messages being less than 30%, the filtering ratio is larger than that of DMLDA with the length of 4; however, it is lower than the filtering ratio of DMLDA with the length of 7. Moreover, the filtering ratio is declining very quickly with the growth of reduplicated proportion, and when the reduplicated proportion is larger than 70%, more than 20% reduplicated messages would be omitted by LEACH. Through
Figure 9 shows the changes of the filtering ratio with different sensor nodes numbers. In our experiment, the proportion of the reduplicated messages is fixed at 50%, and the filtering ratio is tested under three situations: the longest length of the list is 4 in DMLDA, the longest length of the list is 7 in DMLDA, and the delaying cycle is 7 in LEACH. When the list length is set as 4, the slope of the filtering ratio is relatively stable. The filtering ratio is declined steadily with the increase of nodes number, and when there are 100 nodes in WSNs, the filtering ratio has the highest value of 93.3%. If the list length is 7, the change trend of filtering ratio is more complex: the filtering ratio rises from 93.5% to 95.5% when the node number is changed from 100 to 200; however, the filtering ratio remains around 95% when the nodes number is changed from 200 to 300, and when the node number is larger than 300, the filtering ratio begins to decline in a little slope. In LEACH, when nodes number is less than 200, the filtering ratio can be maintained above 90% and a little higher than that of DMLDA with the list length of 4; after nodes number increases to 250, the filtering ratio falls sharply and much lower than that of DMLDA. This experiment shows that the number of the clusters decreases with the reduction of the nodes in WSNs, which leads to fewer filtering nodes to be used. In DMLDA longer message list means fewer filtering nodes along the transmitting route, and hence the filtering ratio does not show obvious advantage comparing with shorter list. In addition, the filtering nodes in LEACH have big time cost to filter messages, so it has advantage in small WSNs. If the number of the nodes in WSNs increases, the number of the clusters and filtering nodes will increase. In DMLDA longer message list means higher capability of filtering and similar number of filtering nodes along transmitting route. However, in LEACH, more nodes mean more clusters, but it cannot dynamically adjust the cluster distribution according to transmission, which leads to the filtering ratio fall sharply. According to the results, it can be concluded that DMLDA is more efficient in larger scale WSNs, especially for the WSNs with longer message list.

This experiment, it can be concluded that the length of the message list affects the filtering ratio obviously, and when the list is longer, the influence of reduplicated messages proportion is weaker. In this experiment, the time cost of LEACH is no less than the memory cost of DMLDA. In particular, DMLDA outperforms LEACH obviously under high reduplicated proportion situation.
node. Conclusively, DMLDA would not bring much extra hardware burden to WSNs with high reduplicated message filtering ratio, but LEACH would bring much time cost to achieve the same results.

4.2. Analysis. According to the simulated results introduced in Section 4.1, DMLDA can maintain high filtering ratio with relatively low extra hardware cost, and the filtering mode based on message list ensures its real time performance. Furthermore, it has obvious advantage in large scale WSNs. The innovation of DMLDA is that a dynamic list is deployed in every cluster head, which can be used to store the history messages to judge whether current message is reduplicated or not. In any filtering node, supposing the message list length is $l$, the omission ratio can be calculated as (6), in which $p$ is the probability of a message being reduplicated. According to (6), assuming a message $m$ is reduplicated, only under the condition that there are $l + 1$ different messages to be transmitted before $m$, and then the $m$ comes to filter node $n$, $m$ may be omitted by $n$. For the distributed structure, there would be many reduplicated messages in WSNs, which lead the value of $p$ larger than it is in the classic networks, and the omission ratio will be much less with relative short list:

$$\alpha = (1 - p)^{l+1}.$$  

The omission ratio in single node can be calculated by (6). In DMLDA, one message would be filtered by several rounds of filtering nodes for clustering-based structure, and its omission ratio would be much less than the result of (6). So when there are more nodes in WSNs, there would be more levels and longer list in filtering nodes, and DMLDA can obtain better effect.

5. Conclusion

In this paper, a new data aggregation scheme named DMLDA is proposed based on clustering routing algorithm, and the whole procedure of DMLDA is described in detail. The main innovations of this paper are summarized as follows: firstly, a special list structure is defined to store history messages, which is used to judge the message redundancy instead of period delay, and this can improve the real time performance of transmitting significantly; secondly, the distribution and initialization of dynamical list have been optimized based on our previous research on clustering-based routing; finally, the content of every list item and list length would be adjusted dynamically with transmission according to filtering state, which ensures high filtering efficiency with little extra hardware burdens for the sensor nodes. Based on these innovations, DMLDA has obvious advantages in large distributed WSNs with many reduplicated messages applications. In the end of this paper, a series of experiments are designed and simulated to prove the performance of DMLDA.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.
[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, pp. 393–422, 2002.
[3] J. Vick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” Computer Networks, vol. 52, no. 12, pp. 2292–2330, 2008.
[4] A. Z. Abbasi, N. Islam, and Z. A. Shaikh, “A review of wireless sensors and networks’ applications in agriculture,” Computer Standards & Interfaces, vol. 36, no. 2, pp. 263–270, 2014.
[5] L. M. L. Oliveira and J. J. C. Rodrigues, “Wireless sensor networks: a survey on environmental monitoring,” Journal of Communications, vol. 6, no. 2, pp. 143–151, 2011.
[6] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, “Data aggregation and routing in wireless sensor networks: optimal and heuristic algorithms,” Computer Networks, vol. 53, no. 7, pp. 945–960, 2009.
[7] T. Du, S. H. Qu, and Q. Wang, “An energy aware ladder diffusion routing algorithm for WSNs,” Lecture Notes on Information Theory, vol. 2, no. 1, pp. 1–6, 2014.
[8] T. Du, S. H. Qu, F. A. Liu, and Q. Wang, “An energy efficiency semi-static routing algorithm for WSNs based on HAC clustering method,” Information Fusion, vol. 21, no. 1, pp. 18–29, 2015.
[9] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom ’00), pp. 56–67, ACM, Boston, Mass, USA, August 2000.
[10] J.-H. Ho, H.-C. Shih, B.-Y. Liao, and S.-C. Chu, “A ladder diffusion algorithm using ant colony optimization for wireless sensor networks,” Information Sciences, vol. 192, pp. 204–212, 2012.
[11] H. Li, K. Lin, and K. Li, “Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks,” Computer Communications, vol. 34, no. 4, pp. 591–597, 2011.
[12] L. A. Villas, A. Boukerche, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, “A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks,” Ad Hoc Networks, vol. 11, pp. 196–209, 2011.
[13] D. Chen and P. K. Varshney, “A survey of void handling techniques for geographic routing in wireless networks,” IEEE Communications Surveys and Tutorials, vol. 9, no. 1, pp. 50–67, 2007.
[14] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, 2002.
[15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, pp. 10–17, Wailea Maui, Hawaii, USA, January 2000.
[16] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” IEEE Transactions on Mobile Computing, vol. 3, no. 4, pp. 366–379, 2004.
[17] M. Bagaa, Y. Challal, A. Ouadjaout, N. Lasla, and N. Badache, “Efficient data aggregation with in-network integrity control for...
[18] C. M. Chao and T. Y. Hsiao, "Design of structure-free and energy-balanced data aggregation in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 229–239, 2014.

[19] H. Yousefi, M. H. Yeganeh, N. Alinaghipour, and A. Movaghar, "Structure-free real-time data aggregation in wireless sensor networks," *Computer Communications*, vol. 35, no. 9, pp. 1132–1140, 2012.
