

Efficient Clustering Routing Algorithm Based on Opportunistic Routing

Fan Wang¹, Yan Yang², Kunpeng Wang¹, Xiaopeng Hu¹, Ning Zhang¹

¹School of Computer Science and Technology, Dalian University of Technology, Dalian, China

²School of Computer and Information Technology, Liaoning Normal University, Dalian, China

Email: wangfan@dlut.edu.cn, xphu@dlut.edu.cn

Received 12 April 2016; accepted 24 May 2016; published 30 May 2016

Abstract

Based on the analysis of the existing classic clustering routing algorithm HEED, this paper proposes an efficient dynamic clustering routing algorithm ED-HEED. In the cluster selection process, in order to optimize the network topology and select more proper nodes as the cluster head, the proposed clustering algorithm considers the shortest path prediction of the node to the destination sink and the congestion situation. In the data transmission procedure, the high-efficiency CEDOR opportunistic routing algorithm is applied into the ED-HEED as the data transmission mode between cluster headers. A novel adaptive dynamic clustering mechanism is also considered into the algorithm, as well as the data redundancy and security control. Our Simulation demonstrates that the ED-HEED algorithm can reduce the energy consumption, prolong the network life and keep the security and availability of the network compared with the HEED algorithm.

Keywords

Wireless Sensor Network, Opportunistic Routing, Clustering Routing Algorithm, Congestion Control

1. Introduction

With the rapid development of mobile communication, embedded computing and sensing technology, wireless sensor network is considered as a more convenient and economical solution in many scopes [1]. Network topology control is one of the core techniques in wireless sensor network and has always been a research hotspot. Every routing protocol is designed on basis of a specific network topology. Therefore, it's important to organize a better network topology for wireless sensor network [2].

Clustering routing technique have lots of advantages and is becoming the major content in the research of wireless sensor network [3]. It can easily control the network topology. Clustering can also conserve the communication cost and make it convenient for intra-cluster data aggregation. In a clustering algorithm, the network is grouped into a set of clusters. Each cluster has a leader, often referred to as the cluster-head (CH), and many member nodes. However, in a special situation, there is only one single node (CH) in a cluster. Nodes in one cluster only communicate with their respective cluster head. They are in charge of collecting data, transmitting

data, and receiving the command from the cluster head. Accordingly, the cluster head is responsible for: 1) Analysis and aggregation of received data. 2) Transmission of the processing result to the sink. In the work [3], researchers prove that clustering algorithm can solve the energy consumption problem better and extend the network lifetime effectively.

LEACH (Low-Energy Adaptive Clustering Hierarchy) [4] is the earliest clustering algorithms in wireless sensor network. Clusters in the network are formed periodically. Cluster head are randomly selected in order to distribute the energy load among the sensors and prolong the network lifetime. In the later research, a number of efficient algorithms are presented based on the idea of LEACH, such as TEEN (Threshold Sensitive Energy Efficient Sensor Network Protocol) [5], DCHS (Deterministic Cluster-head Selection) [6], HEED (Hybrid Energy-efficient Distributed Clustering) [7], etc.

An energy-efficient routing protocol using message success rate in wireless sensor networks is proposed in [8]. Its main idea is to reduce the communication overhead and guarantee the data communication reliability. The algorithm is based on node connectivity and message success rate. However, the complexity of the protocol is higher. In [9], the authors present an energy-efficient and flexible approach using hierarchical agglomerative clustering in wireless sensor networks (DHAC). The protocol only considers one-hop node information to compute the resemblance matrix. The similar nodes are divided into one small cluster. Then, some small clusters are merging into one cluster based on the predefined cluster size threshold. Cluster heads are selected in these clusters. The problem is that the network traffic load is not considered. There still exists unnecessary energy consumption during communication.

In these above algorithms, the network topology is reasonable and cluster heads are well distributed. However, the network performance can still be improved. In this paper, we propose an efficiency dynamic clustering routing algorithm (ED-HEED). We improve HEED protocol including cluster head selection, inter-cluster routing method, re-clustering mechanism, data redundancy and network security control. Compared with HEED protocol, our algorithm reduces the energy consumption, prolongs the network life and keeps the security and validity of the network.

This paper is organized as follows. Section 1 is the introduction. In Section 2, we present the existing problem of HEED and our improvement. Section 3 describes the details of the clustering algorithm (ED-HEED). Section 4 discusses the simulation results and evaluates the algorithm performance. We give the conclusion in section 5.

2. Problem Analysis

2.1. HEED Routing Protocol

HEED (Hybrid Energy-efficient Distributed Clustering) [7] protocol is proposed by Younis O and Fahmy S in 2004. During the cluster head selection, the algorithm considers the residual energy as the primary parameter and the intra-cluster communication cost as the secondary parameter. The node with more residual energy has a larger probability to be selected as a cluster head temporarily. Then the average minimum reachability power (AMRP) between the candidate and all its neighbors is calculated to measure the communication cost. Thus, the cluster heads are selected more properly and the energy within a cluster is more balanced. The HEED protocol has the following features: a distributed method to select the cluster heads; the cluster head selection phase is within limited iterations; the cluster heads distribution is balanced. The HEED protocol performs faster during the clustering. At the same time, the network topology is reasonable and cluster heads are well distributed.

Problems are still existed in HEED protocol. When a node locates in an area within the range of several different clusters, it will choose a cluster head with the minimal average power. In a practical application, this method may ignore some factors such as the distance between a node and its cluster head, the distance between cluster heads and the sink node and so on. Large amounts of nodes join a cluster within which its cluster head has a long distance from the sink node. So the communication cost of the cluster head is too high. Besides, network traffic also plays an important role in the algorithm.

2.2. Improvement of HEED

To overcome the above shortcomings, we improve the HEED protocol by proposing an efficiency dynamic clustering routing algorithm (ED-HEED). During the cluster head selection, we comprehensively consider the residual energy, shortest distance from the node to the sink node and the network traffic congestion. So the clus-

ter heads are selected more properly. Then, we use an opportunistic routing protocol for inter-cluster data transmission. We also present a novel adaptive dynamic re-clustering algorithm, the network security control and data redundancy control mechanism.

3. ED-HEED Clustering Routing Algorithm

ED-HEED is an efficient dynamic clustering routing algorithm. It is designed with the following ideas. 1) In cluster formation, we consider not only the node residual energy, but also the shortest distance prediction and node traffic congestion. 2) During data transmission phase, cluster member nodes communicate with the cluster head directly, but multi-hop cluster head to cluster head connectivity uses an opportunistic routing protocol (CEDOR [10]).

In order to avoid the fast energy consumption of the cluster heads, we adopt an adaptive dynamic re-clustering mechanism to our ED-HEED. The network will re-clustering when the Sink real-time detects the energy of some cluster head has dropped below the predefined threshold or the node is invalidate. The strategy will indeed improve the network performance.

According to the idea that nodes broadcast HELLO frames to probe network link in traditional dynamic routing protocol (AODV), we use the similar way to detect the working state of each node. In our algorithm, each node periodically broadcasts HELLO frames to all its neighbors to find their working state. Then, we can ensure the security and validity of the network. At the same time, cluster head need broadcast three thresholds to all member nodes for further reducing the redundant data and energy consumption of the member nodes. Actually, we use the simple HELLO frame mentioned above to carry the thresholds value, which will minimize the control message.

3.1. Cluster Formation Algorithm

In order to reduce energy consumption more efficiently and guarantee the network security and validity, we consider the shortest distance from node to the base station (the sink node) and traffic congestion as another two parameters during cluster head selection. The probability of becoming a cluster head is modified as follows.

$$CH_{prob} = C_{prob} \times \left(\alpha \times \frac{E_{Residual}}{E_{Max}} + \beta \times DP_i + \gamma \times C_i \right) \quad (1)$$

$$\alpha + \beta + \gamma = 1$$

where α , β and γ are the weights of the current residual energy, shortest distance and node congestion. They can set to different value according to specific situation. In this paper, we don't discuss it much and just make them satisfied the equation $\alpha = \beta = \gamma = 1/3$. $E_{Residual}$ is the estimated current residual energy of the node and E_{Max} is a reference maximum energy. DP_i is the prediction of the shortest distance from the node to the sink node.

ED-HEED algorithm uses the multiple priority queue model described in [10] when computed the node congestion. However, we only consider the high priority queue while computing the traffic congestion of the node. The congestion factor, C_i is described as follows.

$$C_i = \frac{\sum_{i=1}^{N_H} LH_i}{Buf_H} \quad (2)$$

where N_H is the number of packet in the current high priority queue. LH_i is packet length (bit) of the high priority queue. Buf_H is the buffer size (bit) of the high priority queue.

3.2. Inter-Cluster Routing Protocol

In the work [10], the algorithm CEDOR (Congestion Control Based on Multi-priority Data for Opportunistic Routing) was proposed. It comprehensively considers the shortest distance from the node to the Sink, the current residual energy and traffic congestion of the node. So we apply CEDOR as the inter-cluster routing protocol. The network security, effectiveness and the data transmission efficiency are all improved.

When we apply CEDOR routing algorithm, there's some work we need to do. Firstly, we define the neigh-

bours set of each node, S_{nbr} . Secondly, we compute the delivery utility (DU_i) of each cluster head to find candidate forward node set. Then cluster head broadcasts data to each node in the candidate forward node set. Finally, data is transmitted to the Sink through multi-hop. So after cluster formation phase, each cluster head knows its cluster head neighbour node set and its own depth. During data transmission, the node current congestion is obtained by computing the priority of forwarding data. The pseudo code of cluster head routing algorithm is shown in **Figure 1**.

3.3. Re-Clustering Mechanism

In classical clustering routing algorithms (e.g. LEACH, HEED), clustering occurs after a fixed period. The period is predefined before running the algorithm. However, selecting an appropriate period will directly affect the performance of clustering routing algorithm [11]. The cluster head will die for fast energy consumption if the period is too long. And if the period is too short, frequent clustering will affect the efficiency of data transmission.

To overcome the shortcomings, a novel adaptive dynamic re-clustering mechanism is proposed. Cluster head records its current residual energy (E_{Record}) before stable data transmission. Cluster heads are real-time monitoring its current energy and compares it with αE_{Record} (we assume that $\alpha = 0.5$). If the current residual energy is less than that, cluster head will send an emergency high priority packet to the Sink which means sending a request for re-clustering. On the other hand, the network also starts re-clustering if some cluster node is found invalidate or dead (see Chapter 3.4). The pseudo code of the re-clustering mechanism algorithm is shown in **Figure 2**.

Algorithm 5: Cluster Header Routing Algorithm.

```

1 begin
2   Initial the cluster neighbor node set  $\rightarrow S_{nbr\_CH}$ 
3   if ( $S_{nbr\_CH} == \emptyset$ )
4     Select the Sink as the next hop
5   else
6     Compute the delivery utility of each cluster header in  $S_{nbr\_CH}$ 
7      $DU_i = W_1 \times C_i + W_2 \times (E_{init} - E_{left})/E_{init} + W_3 \times D_i/D_{max}$ 
8     Compute the expected value of the delivery utility
9      $\overline{DU} = \frac{\sum_{j=1}^N DU_{ij}}{N}$ 
10    for (each  $V_i \in S_{nbr\_CH}$ )
11      if ( $DU_i < \overline{DU}$ )
12        set priority of  $V_i$  according to the  $DU_i$ 
13         $CN\_CH_i = CN\_CH_i \cup V_i$ 
14      endif
15    endfor
16    // Broadcast packets to the node in  $CN\_CH_i$ 
17    Sendbroadcast(packet,  $CN\_CH_i$ )
18    for (each  $V_i \in S_{nbr\_CH}$ )
19      if ( $V_i$ 's priority is the highest)
20        // Broadcast the ACK to the other nodes in  $CN\_CH_i$ 
21        Sendbroadcast(ACK,  $CN\_CH_i$ )
22        break
23      else
24        // Wait the ACK from the nodes whose priority is higher than  $V_i$ 
25        Wait ( $CN\_CH_i$ )
26        if (The node does not receive any packet in a period)
27          // Broadcast the ACK to the other nodes in  $CN\_CH_i$ 
28          Sendbroadcast(ACK,  $CN\_CH_i$ )
29          break
30        endif
31      endif
32    endfor
33  endif
34 end

```

Figure 1. The cluster header selection pseudo-code.

Algorithm 6: The Re-clustering Mechanism Algorithm.

```

1 begin
2   // Record current residual energy locally for every cluster header nodes
3    $E_{Record} = E_{Left}$ 
4   for (each  $V_i \in SCH$ )
5     if ( $V_i == Sink$ )
6       // Receive message, and judge whether it is equal to cluster
7       GetMessage (msg)
8       if (msg == cluster)
9         // Sink broadcast synchronization cluster signal to the network
10        Sendbroadcast (cluster)
11      endif
12    else
13      if ( $E_{Left} \leq E_{Record}/2$ )
14        //Request for re-clustering
15        SendMessage(cluster, high-priority, Sink)
16      endif
17    endif
18  endfor
19 end

```

Figure 2. The re-clustering mechanism pseudo-code.

3.4. Network Security and Redundancy Control

In wireless sensor network application, nodes may suddenly fail or die for some reason. We need to know their working state real-timely, so we can take measures to make sure the network is validate and secure. Depending on the idea that nodes broadcast HELLO frames to probe network link in traditional dynamic routing protocol (AODV), we use the similar way to resolve the problem.

Each node of the network maintains a neighbour connection list and a timer. The neighbour connection list describes the node connectivity with all its neighbours. We assume each node exchanges information with all its neighbours directly. We define a simple HELLO frame and let $TTL = 1$. Each node broadcasts it to all its neighbour after a fixed period, HELLO_INTERVAL. If a node does receive the HELLO frame from its neighbour, the connectivity between them is certainly normal and the connectivity information will be recorded in its neighbour connection list. The list will update during the running of the network. At last, each node knows the connectivity situation with all its neighbours.

However, if the node doesn't receive any data or the HELLO frame from one of its neighbours after a limited time, $ALLOWED_HELLO_LOSS * HELLO_INTERVAL$, it's possible that the neighbour node is failed. Then it will immediately send an emergency high priority packet to the sink. Receiving the packet, the sink will take measures to handle it.

After the cluster formation phase, the sink will generate three thresholds (T_{hard1} , T_{hard2} , T_{soft}) depending on the effective range of the current received sensor data. Then the thresholds will be broadcasted to every node of the network. Where T_{hard1} is the lower bound of the effective range. T_{hard2} is the upper bound of the effective range. T_{soft} is the maximum variation range of the sensor data. In general, the threshold information is carried in the above HELLO frame. So extra communication cost is reduced effectively.

During the stable data transmission phase, the first data collected by each node is transmitted to the cluster head respectively and recorded locally as msg_record . After that, each node will compare the current data with the thresholds as follows: the node will send raw data under the condition that the data message is beyond the scope of T_{hard1} to T_{hard2} or the difference between the data and the msg_record is bigger than T_{soft} . Otherwise, the node will simply transmits a control packet to cluster head instead of the sensor data. Thus, not only the network function well but also the total number of transmission and reception data and the energy consumption of cluster member node is reduced effectively. However, there's still a serious problem existed which is that if the cluster head didn't receive the data from some member node, it's hard to tell the member node is failed or doesn't need to send data temporarily. To solve this problem, the network security mechanism presented above

is applied to failure detection in our ED-HEED algorithm. So the network is running safely and reliably. **Figure 3** describes the data redundancy control algorithm.

4. Simulation and Performance Evaluation

4.1. Simulation Setup

The ED-HEED algorithm proposed in this paper is experimented in a simulation platform based on the NS-2 software. We set up our simulation parameters as shown in **Table 1**. 150 nodes are randomly deployed in a rectangle region of 25 m \times 25 m. There is only one base station which is located in the center of the network. The result of the deployment is shown in **Figure 4**. The little red circle is the normal node and the blue one represents the sink in the figure.

Algorithm 7: Data Redundancy Control Algorithm.

```

1 begin
2   // Record data locally
3   msg_record = message
4   // Collect data for the first time
5   if (firts)
6     SendMessage (message,CH)
7   else
8     // Data is beyond the scope of the effective threshold
9     if (message <  $T_{Hard1}$  || message >  $T_{Hard2}$  )
10      SendMessage (message,CH)
11     // Data fluctuation is too big than  $T_{Soft}$ 
12     elseif (abs(message - msg_record >  $T_{Soft}$ ))
13      SendMessage (message,CH)
14     else
15       sleep()
16     endif
17   endif
18 end

```

Figure 3. Data redundancy control pseudo-code.

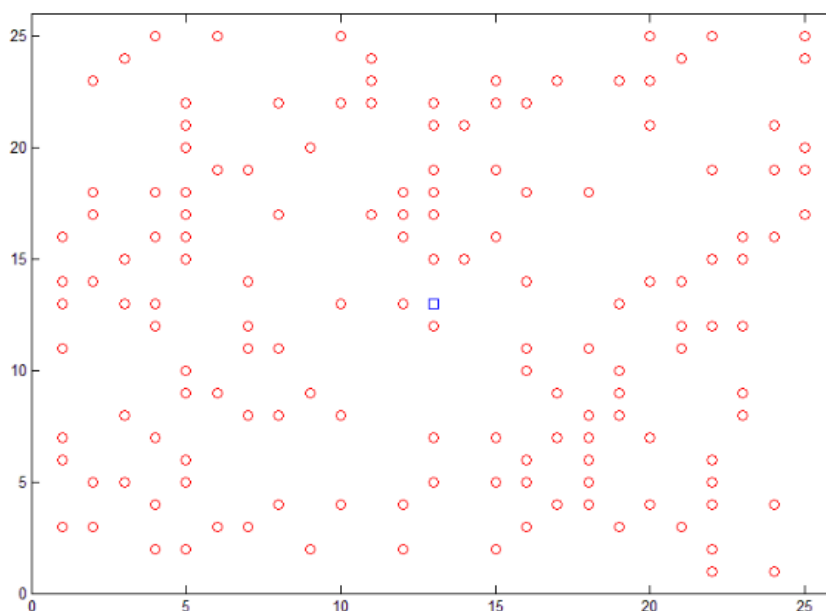


Figure 4. The random deployment of all nodes in the network.

Table 1. Parameter of the experiment.

Parameter	Value
Network Size	(0, 0) - (25, 25)
Number of Nodes	150
Coordinates of the Sink	(12, 12)
Initial Energy	50 J
Energy Consumption for Processing One Bit [12]	0.01 J/bit
Amplifier Parameter of Free Space Model [12]	0.01 J/bit/m ²
Amplifier Parameter of Multipath Model [12]	0.0013 J/bit/m ²
Size of Control Packet	64 bit
Size of Data Packet	500 bit

We provide comparison experiments between our ED-HEED clustering algorithm and the HEED algorithm in some aspects of the network performance.

4.2. The Comparison of Network Lifetime

Network lifetime has always been one of the most important criteria to evaluate the network performance. The experiment is aiming at the comparison of network lifetime using different algorithms in terms of the number of alive nodes. Our experiments collected the statistical information of the network after a fixed period instead of each round. The energy of each node was monitored continuously. When its initial energy was lost, a node would stop working.

Figure 5 depicts that the number of alive nodes changes over time. From the figure, we can see some nodes had lost all of its energy when time reached 200 s in HEED algorithm. But the phenomenon began at 350 s by our method. ED-HEED algorithm we proposed improves HEED in terms of the cluster formation, inter-cluster transmission, data redundancy control and so on. Therefore, the nodes survive one time longer. The network lifetime is extended effectively.

4.3. The Comparison of Energy Balance

We ran HEED and ED-HEED algorithm under the same experimental condition. Once some node died, the residual energy of each node was computed. According to the statistic of the data, we got the distribution of nodes' residual energy which is shown in **Figure 6**. Where X-axis means the different distributed range of the residual energy and Y-axis is the number of nodes.

The experiment results can measure the energy balance problem. From **Figure 6**, we can see that half of the nodes in the network have less than 30% of the maximum energy when HEED was adopted. Whereas, there're more than 75% of the nodes in the network in ED-HEED. So our algorithm can achieve energy balance more effectively.

4.4. The Comparison of Re-Clustering Time Interval and Data Transmission Rate

In this research, we assume that the fixed period between each round is 50 s in HEED. Each re-clustering round schedule is shown in **Table 2** as follows.

In ED-HEED, the cluster head won't be reselected until the energy of some cluster head has dropped down for a half. From **Table 2**, we can draw a conclusion that all nodes were full of energy at the beginning so that the re-clustering time interval was longer. Then the stable data transmission would certainly last for a relative longer time. With time went on, the re-clustering time interval was getting shorter. Thus, the energy consumption is reduced.

Through the above analysis, we know ED-HEED can extend network lifetime effectively and improve the reliability and security. Now we compare the data transmission rate between ED-HEED and HEED. We count the number of packet successfully received within a fixed time to represent the network transmission efficiency. The

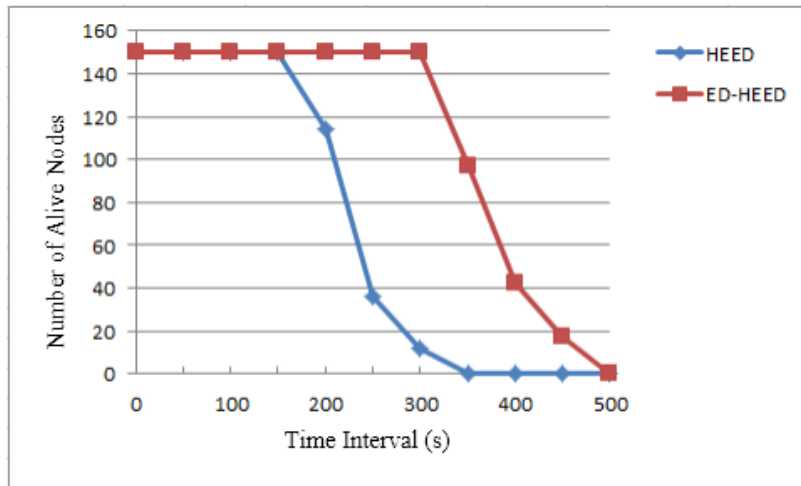


Figure 5. Number of alive nodes diagram.

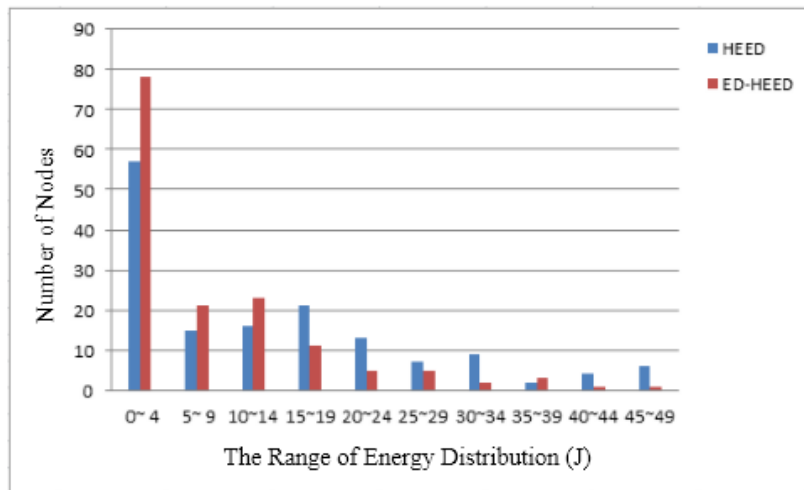


Figure 6. The comparison of residual energy distribution.

Table 2. Each re-clustering round schedule table.

Round	1	2	3	4	5	6
HEED	50 s	100 s	150 s	200 s	250 s	300 s
ED-HEED	154 s	268 s	334 s	391 s	428 s	456 s

result is shown in Figure 7.

4.5. The Comparison of Transmission Latency

We carry on this experiment to evaluate that ED-HEED algorithm can effectively reduce network transmission latency. So that emergency high priority packets can be sent to the sink as soon as possible. To make the comparison convenient, we randomly selected a cluster head. Let the node send a high priority packet to the sink. Then, we traced the packet and recorded the latency when it arrived at the sink. The result is described in Figure 8.

Figure 8 shows us that ED-HEED has a lower transmission latency than HEED. Therefore, ED-HEED clustering routing algorithm can process the high priority packets perfectly in time. The network security and reliability are guaranteed.

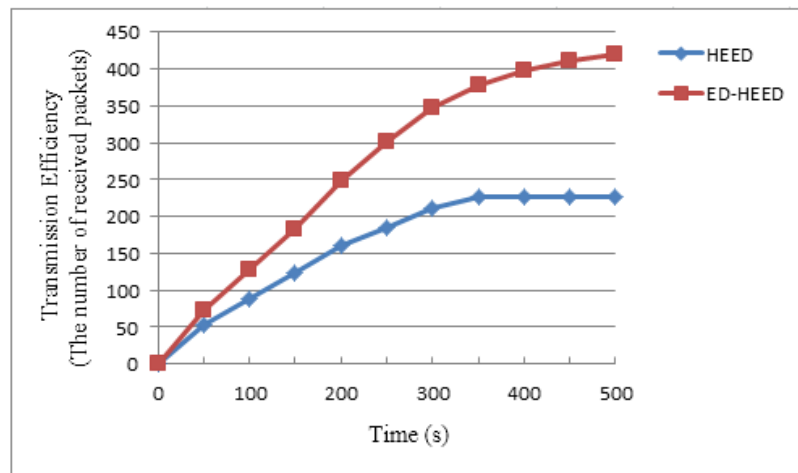


Figure 7. The comparison of transmission efficiency.

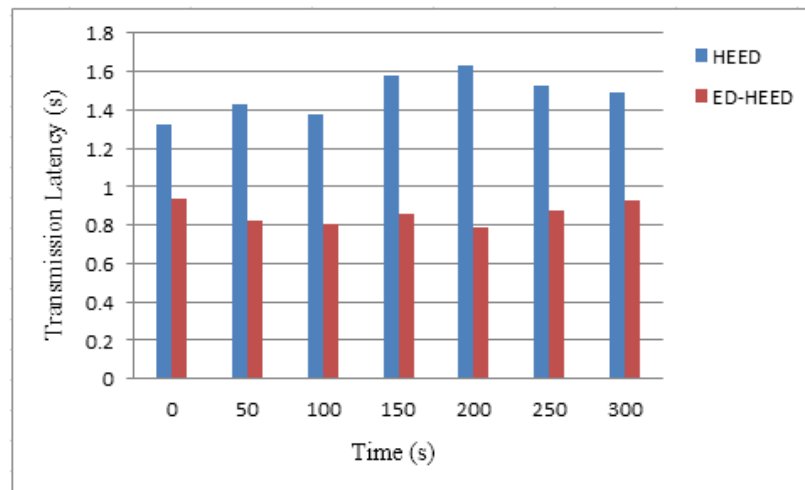


Figure 8. Transmission latency diagram.

4.6. The Comparison of Data Redundancy Control

As is known to all, sensor data collected by a single node are always correlated temporally. We gathered the data transmitted by one node every 10 s to evaluate the validity of data redundancy control mechanism. **Figure 9** is the analysis diagram of data redundancy control.

From the figure, we can see that there's no difference between ED-HEED and HEED when the node sent data to its cluster head the first time. However, after that, the data gathered by the node hardly changed and stayed stable. So there's no need to send the redundant data to its cluster head. In ED-HEED, only a simple control packet was transmitted to inform the cluster head that the node was operating well. Thus, the data volume is apparently reduced.

5. Conclusion

In this paper, we propose an efficiency dynamic clustering routing algorithm ED-HEED based on comparison and analysis of the existing classic clustering routing algorithm HEED. We improve the algorithm as follows: 1) for the purpose of optimizing the network topology and selecting more appropriate cluster head, we take the prediction of shortest path from the node to the Sink and node congestion into consideration during cluster formation phase. 2) In data transmission phase, the high-efficiency CEDOR opportunistic routing algorithm is applied to ED-HEED as the data transmission mode between cluster headers. 3) We also present a novel adaptive

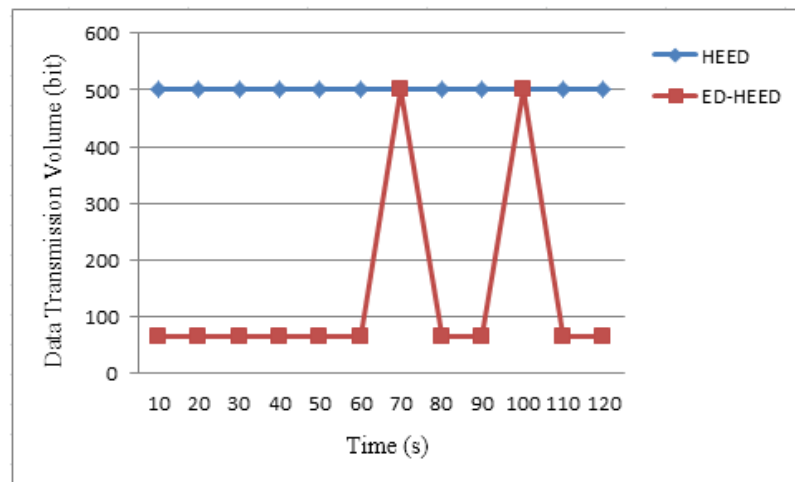


Figure 9. The analysis of data redundancy control.

dynamic re-clustering algorithm, the network security control and data redundancy control mechanism. The simulation results demonstrate that ED-HEED is more effective in reducing network energy consumption and extending network lifetime. The transmission latency of high priority packets is also lower and the network is more secure and reliable.

Acknowledgements

This research was supported by “National Natural Science Foundation of China” (No.61272523), “the National Key Project of Science and Technology of China” (No.2011ZX05039-003-4) and “the Fundamental Research Funds for the Central Universities”.

References

- [1] Ren, F.Y., Huang, H.N. and Lin, C. (2003) Wireless Sensor Networks. *Journal of Software*, **14**, 1282-1291.
- [2] Zhao, T., Guo, T.D. and Yang, W.G. (2009) Energy Balancing Routing Model and Its Algorithm in Wireless Sensor Networks. *Journal of Software*, **20**, 3023-3033. <http://dx.doi.org/10.3724/SP.J.1001.2009.03384>
- [3] Pantazis, N.A., Nikolidakis, S.A. and Vergados, D.D. (2013) Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, **15**, 551-591. <http://dx.doi.org/10.1109/SURV.2012.062612.00084>
- [4] Heinzelman, W.R., Chandrankasan, A. and Balakrishnan, H. (2000) Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Hawaii, 4-7 January 2000, Vol. 2, 10. <http://dx.doi.org/10.1109/hicss.2000.926982>
- [5] Manjeshwar, A. and Agrawal, D.P. (2001) TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. *IPDPS*, **1**, 189. <http://dx.doi.org/10.1109/ipdps.2001.925197>
- [6] Handy, M.J., Haase, M. and Timmermann, D. (2002) Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection. 2002 *IEEE 4th International Workshop on Mobile and Wireless Communications Network*, 368-372. <http://dx.doi.org/10.1109/mwcn.2002.1045790>
- [7] Younis, O. and Fahmy, S. (2004) HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing*, **3**, 366-379. <http://dx.doi.org/10.1109/TMC.2004.41>
- [8] Yoon, M., Kim, Y.K. and Chang, J.W. (2013) An Energy-Efficient Routing Protocol Using Message Success Rate in Wireless Sensor Networks. *Journal of Convergence*, **4**, 15-22.
- [9] Lung, C.H. and Zhou, C. (2010) Using Hierarchical Agglomerative Clustering in Wireless Sensor Networks: An Energy-Efficient and Flexible Approach. *Ad Hoc Networks*, **8**, 328-344. <http://dx.doi.org/10.1016/j.adhoc.2009.09.004>
- [10] Ding, N., Tan, G.-z., You, D. and Zhang, W. (2013) Opportunistic Routing Algorithm Based on Topological Time-Varying and Residual-Energy Balance in Wireless Sensor Network. *Journal of Electronics & Information Technology*, **35**, 715-720.

- [11] Heinzelman, W.B., Chandrakasan, A.P. and Balakrishnan, H. (2002) An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*, **1**, 660-670.
<http://dx.doi.org/10.1109/TWC.2002.804190>
- [12] Rappaport, T.S. (2006) *Wireless Communications: Principles and Practice*. Prentice Hall PTR.