Research Article

The Degree-Constrained Adaptive Algorithm Based on the Data Aggregation Tree

Xiaogang Qi,1 Zhaohui Zhang,1 Lifang Liu,2 and Mande Xie3

1 School of Mathematics and Statistics, Xidian University, Xi’an 710071, China
2 School of Computer Science and Technology, Xidian University, Xi’an 710071, China
3 College of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Correspondence should be addressed to Mande Xie; ximd@zjgsu.edu.cn

Received 29 September 2013; Accepted 25 January 2014; Published 27 February 2014

Academic Editor: Ying-Hong Wang

Copyright © 2014 Xiaogang Qi 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.

In the PEDAP algorithm, a minimum spanning tree considering the energy consumption is established based on the Kruskal algorithm, and updated every 100 rounds. There exists a defect that the energy of some nodes rapidly expires because the degrees of nodes differ significantly, and the delay time is not considered. Based on the above analysis, a new algorithm called DADAT (a Degree-based Adaptive algorithm for Data Aggregation Tree) is proposed. The energy consumption and the delay time are both considered, and a weight model to construct a minimum spanning tree is established. Furthermore, the node degree on the tree is readjusted according to the averaged degree of the network, and nodes are labeled by red, yellow, and green colors according to their remaining energy; the child nodes of the red nodes are adaptively transferred to their neighbor nodes which are labeled as green. Finally, we discuss the weight and the update rounds’ impact on the network lifetime. Experimental results show that the algorithm can effectively balance the energy consumption and prolong the lifetime of the network, as well as achieving a lower latency.

1. Introduction

With the development of wireless communication technology, more and more high-performance, low-cost products are widely used and this significantly promotes the progress of the wireless sensor networks (WSNs). Wireless sensor networks are composed by a large number of stationary or mobile sensors. Its widely used in national defense, environmental monitoring, traffic management, biomedicine, disaster monitoring, manufacturing, gas turbine engine testing, and other fields [1, 2]. Wireless sensor nodes have the characteristics of large number as well as small size, limited communications, and computing ability; the power of nodes’ battery is limited and cannot be charged. In addition, its initial energy is heterogeneous. Therefore, how to utilize energy effectively, to ensure effective data transmission and prolong the network lifetime, will be the key issues in the process of design the network algorithms or protocols.

The sensor nodes in the monitoring region always send its information to sink node by the way of ad hoc network, and the sink node is located at the centre or on the border of the network, so the other nodes must transmit data directly or indirectly to the sink node through single hop or multiple hops. Densely deployed sensor network for data collection and transmission contains large number of redundant data, but the node energy consumption in data process is far less than that in the data transmission. Accordingly, the data aggregation techniques to minimize the energy consumption of data transmission are necessary and helpful to reduce the amount of the transmission data and achieve efficient utilization of energy. The mediate nodes aggregate the received data based on aggregate functions, segment the aggregated data into fixed-size packets, and then transmit to the next hop node [3], and thus the redundant information and unnecessary energy consumption is reduced in communication process according to the data correlation. To solve the problem, the most common method is to build an aggregation tree rooted at the sink [4, 5], in which each mediate node aggregates the data from their children and the data their own sensing information into a fixed-size...
packet and then sends to its parent node. As the tree is the minimal connectivity subgraph structure in a network, the tree-based data collection protocol would ensure network connectivity and reliability with guaranteed QoS effectively and easily implement efficient energy saving and so forth, and the tree-based schemes become a hot one currently. Typical tree-based data collection protocols include EDAET (an energy-aware distributed heuristic) [6], PEDAP (power efficient data gathering and aggregation protocol) [7], MNL (maximum network lifetime) [8], IAA (iterative approximation algorithm) [9], CNS (center at nearest source), SPT (shortest path tree), and GIT (greedy incremental tree) [10]. In practical applications, the amount of sensor nodes is great and the nodes are widely distributed, so the topology control and the data transmission are hard, and some distributed algorithms are difficult to be carried out considering the data aggregation. Sensor nodes independently select the data return path, the path formed by the intersection is random and cannot implement the effective aggregation of sensing data, so it can be further optimized.

In addition, the source nodes have a large number of data packets sent to the sink in case of the network link failure; the time waiting data packet and the different data aggregation computing time lead to an increase in the network delay. In many real-time applications, especially in target tracking, medical care, fire monitoring, and structural health diagnosis, it requires the strict time limits for data packet transmission. Also, the outdated information may lead to a negative impact on detection system, and the data in complex delay-constrained network is required to be transmitted with priority to eliminate the queuing delay on the mediate nodes. Therefore, during the network algorithm design, we have to consider how to establish efficient routing strategies for data aggregation and reduce the energy consumption and network data transmission delay to ensure the real-time application requirements and prolong the network lifetime.

Based on the above analysis, several key factors are considered in our scheme, such as the number of hops, residual energy, aggregation factor, and so on, and then a distributed algorithm based on the degree of adaptive dynamic aggregation tree (DADAT) is proposed. In DADAT, the weight of the node model is established, and then an aggregation tree rooted at the sink is built up based on the Kruskal’s greedy algorithm. In addition, the nonleaf node updates and maintains the tree according to the average degree and its own energy information and ensures a balanced use of energy in each round. The data packets are aggregated in nonleaf node based on aggregation strategy, and this can reduce the amount of data transmitted, so that the node energy is consumed in a more efficient way, and the network lifetime is prolonged and a lower latency is achieved.

2. Network and Communication Model

2.1. Network Model. Assume that $N$ sensor nodes are randomly deployed in the area of size $L \times L$; the nodes send data to the sink node periodically. Sink node is on the edge of the monitoring region and is responsible for receiving collected data for analysis. The wireless sensor network topology can abstract into a two-dimensional plane undirected graph $G(V, E)$, where $V$ and $E$ represent the set of nodes and edges, respectively. $E \subseteq V \times V$, $|V| = N$, represents the number of nodes. The energy of each node initialization is $E_{\text{init}}$, the current residual energy is $E_{\text{current}}$. In order to facilitate model analysis and describe WSNs’ topology, we make the following assumptions:

1. all nodes no longer move after deploying;
2. all nodes in the same structure and with the same initial energy and energy cannot be added;
3. each node has the same communication radius $r$;
4. all nodes know the network location information of all neighbors.

2.2. Communication Model

2.2.1. Energy Consumption of Communication. Our energy model is the same as what is described in the literature [11], the energy consumption of sending data and receiving data according to the wireless communication model is formulated, respectively, as follows, and $E_{\text{da}}$ is the energy consumption while aggregating 1 bit data:

$$
E_{\text{tx}}(l,d) = lE_{\text{elec}} + l\epsilon_{tx} d^2,
$$

$$
E_{\text{rx}}(l) = lE_{\text{elec}},
$$

$$
E_{\text{ag}} = lE_{\text{da}},
$$

where $E_{\text{tx}}(l,d)$ is the energy consumption of receiving $l$ bit data, $E_{\text{rx}}(l)$ is the energy consumption of sending $l$ bits data, $E_{\text{elec}}$ is the energy consumption of wireless transceiver circuit in sending and receiving data, and $\epsilon_{tx}$ is the magnification of the signal amplifier power in the free-space model.

2.2.2. Communication Delay. To save energy, MAC layer communication protocol should work between the sleeping and listening modes. First, all nodes in the network are set in sleeping state, and some will switch to the listening mode when they are awakened and begin aggregating data while listening mode is completed. When the transmitting process is completed, these nodes will switch to sleeping mode again and continue to wait the beginning of the next listening period. Therefore, the delay for the node is calculated as

$$
\text{Delay}(i) = D_s(i) + D_l(i) + D_{tx}(i) + D_{ag}(i),
$$

where $D_s(i)$ is the $i$th node’s sleeping time, $D_l(i)$ is the listening time, and $D_{tx}(i)$ and $D_{ag}(i)$ represent the time for the $i$th node’s transmitting and aggregating the data packets,
respectively. The $D_{tr}(i)$ and $D_{ag}(i)$ are calculated according to the following formulas:

$$D_{tr}(i) = \frac{\text{Packet}(i)_{num_{tr}} \times \text{Packet size}}{B},$$

$$D_{ag}(i) = \frac{\text{Packet}(i)_{num_{ag}} \times \text{Packet size}}{\phi},$$

where $\text{Packet}_{num_{tr}}$ and $\text{Packet}_{num_{ag}}$ are the number of data packets before and after aggregation for the node $i$, respectively, $\text{Packet size}$ is the size of a data packet, $B$ is the channel width, and $\phi$ represents the time aggregate unit packet. Accordingly, the average network delay is calculated as

$$\overline{D} = \frac{1}{N} \sum_{1}^{N} \text{Delay}(i),$$

where $\overline{D}$ is the average network delay, $\text{Delay}(i)$ is the delay for the $i$th node, and $N$ is the number of the nodes.

2.2.3. Aggregation Strategy. In the PEDAP, the $i$th node aggregates its own originated packets and children’s data packets into a new data packet and then transfers it out. However, the sink node requires a certain amount of original data requirements in some networks [12], but the amount of data of a lot of nodes to the sink node is little based on above aggregation algorithm. This will result that the accuracy of the data cannot be guaranteed, so the original data from source nodes to the sink should be as much as possible. We assume that the parent of source node $i$ is $j$, the parent of node $j$ is $k$, and node $k$ is out of the communication radius of node $i$, so correlation between nodes $i$ and $k$ is relatively low, and we hope that a small amount of original data of node $i$ is aggregated and more of original data is reserved on node $k$. For example, there just exists 25 percent data originated from its child node $i$ on node $k$ while the amount of the data decreases by about 50% after each aggregation; the original information to sink node through multiple hops will be very little or almost all lost. So there exist fatal flaws for these types of data aggregation algorithms. To solve the above flaws, we redesign the aggregation policy as follows, and the definition of node aggregation coefficient $\rho(i)$ is calculated

$$\rho(i) = \frac{\text{hop}(i) - 1}{\text{hop}_{\max}},$$

where $\text{hop}_{\max}$ is the maximum number of hops from leaf nodes to sink on its branch, and $\text{hop}(i)$ is the hop numbers from node $i$ to the sink.

Therefore, there exists a relationship between the original data packet and the aggregated data packet for the node $i$ as listed in (6), where $\rho(i)$ is the data aggregation coefficient on node $i$. This is significant in the aggregation mode and leads to achieve a better data collection of accurate network information:

$$\text{Packet}(i)_{num_{ag}} = \text{Packet}(i)_{num_{tr}} \times (1 - \rho(i)).$$

3. Data Aggregation Based Spanning Tree Algorithm

In GIT algorithm, the distance between each pair nodes is taken into account to build a tree which starts from the sink node, and its performance is much better. However, it needs the global information and is difficult to be achieved in the distributed pattern. In PEDAP algorithm, it is a near optimal minimum spanning tree-based routing scheme, and the tree structure is updated for each 100 rounds. Because it does not take into account the residual energy of nodes, the minimal energy nodes are easily to be dead. PEDAP-PA uses the information about the residual energy of each node and considers the balance of energy consumption.

3.1. Weight Model. To solve the problem, we select the numbers of hops, the residual energy, and aggregation coefficient to redesign the data collection scheme to prolong the network’s survival time while ensuring data accuracy. The weight model is formulated as follows:

$$W = 1 - \frac{\alpha \times \frac{\text{hop}_{\text{pre}}}{\text{hop}_{\text{pre}}(i)} + (1 - \alpha) \times \frac{E(i)_{\text{current}}}{E_{\text{init}}}}{\text{hop}_{\max}}.$$  (7)

Based on (7) and (5), we have

$$W = \frac{\text{hop}_{\max}}{\text{hop}_{\text{pre}}(i) - 1} \left[\alpha \times \frac{\text{hop}_{\text{pre}}}{\text{hop}_{\text{pre}}(i)} + (1 - \alpha) \times \frac{E(i)_{\text{current}}}{E_{\text{init}}}\right]$$

$$= \alpha \times \frac{\text{hop}_{\max} \times \text{hop}_{\text{pre}}}{(\text{hop}_{\text{pre}}(i) - 1) \times \text{hop}_{\text{pre}}(i)} + (1 - \alpha) \times \frac{E(i)_{\text{current}}}{E_{\text{init}}},$$  (8)

where $\text{hop}_{\text{pre}}(i)$ represents the prehop of node $i$, $\text{hop}_{\text{pre}}(i) = \lceil \text{dist}(i, \sin k)/r \rceil + 1$, $\text{dist}(i, \sin k)$ is the distance from node $i$ to sink node. $\text{hop}_{\max}$ is the maximum of prehop, $\text{hop}_{\text{pre}}(i)$ is the average prehop, $E(i)_{\text{current}}$ is current energy, and $E_{\text{init}}$ is the initial energy. Because the network is homogeneous, here the initial energy of each node is $E_{\text{init}}$. In Section 4, we will discuss the weight of $\alpha$ in detail.

3.2. The Establishment of the Tree and Adaptive Maintenance. In Section 3.1, we create a weight model, establish a tree starting from the sink node, and select the maximum weight node to be added to the current tree until all nodes in the network are added to the tree, which is reconstructed after a certain number of rounds.

In the process of establishing tree, we take into account the hops, residual energy, and amount of data, but the load of the node cannot be balanced. Some nodes have more children while others have few; this may lead to sharp energy consumption on some nodes. To solve this subproblem, we have to reorganize some nodes in the network. First, we compute the average degree of nonleaf nodes and let the
Figure 1: The node transfer operation.

Figure 2: The transfer diagram among red nodes, yellow nodes, and green nodes.

According to the above method, some nodes consume more energy than others do in the network because the nodes in different layers consume energy differently. When the residual energy of these nodes is not enough to meet the energy requirements of the transmission of data, this will result in the loss of the data packets. To solve this problem, we can transfer the children of these nodes to the node with more residual energy in the same layer to prolong the lifetime of the network. The node is labeled in red color if the node’s residual energy is less than a given value $E_{\text{low}}$, the node is labeled in green color if its residual energy is more than another certain value $E_{\text{high}}$, and the remaining nodes are to be set in yellow color.

In Figure 2(a), node 3 is a red node, node 4 is a yellow node, others are green nodes, and node 3 will die soon if the network topology control does not change in time. For node 3, node 2 and node 4 become the potential parents for the children of node 3. It is noted that node 4 is yellow already but node 2 is green, so node 2 becomes the parent of node 3. Therefore, this can well balance the energy consumption for node 3 and its neighbor nodes and prolong the network’s lifetime.

Such a tree is adaptively revised based on average degree to make the tree maintain a better structure. It is helpful to prolong the network lifetime while ensuring the data latency and data volume requirements. Algorithm for the specific process is listed as follows.

**Step 1.** Set sink as the root, then apply the Kruskal greedy algorithm, select the node which has the largest value based on formula 11, and join the tree firstly; turn to Step 2.

**Step 2.** Calculate the averagedegree of nonleafnode, transfer the node based on it average degree of the node, and then calculate the degree of each node; turn to Step 3.

**Step 3.** Calculate the residualenergy of each node; if the number of dead nodes reached a certain number, the algorithm stops; otherwise divide all nodes according to the residual energy into red, yellow, and green nodes. And then transfer the red node’s children to the green nodes in the same layer, and calculate the degree of each node; turn to Step 4.

**Step 4.** When the number of rounds to meet certain conditions, turn to Step 1, rebuild the new aggregation tree, or turn to Step 2.

And the algorithm flowchart is shown as in Figure 3.
Table 1: Experimental parameters.

| Parameter                      | Value              |
|--------------------------------|--------------------|
| Network distribution field     | 300 m × 300 m      |
| Node number                    | 300                |
| Sensing range (m)              | 32                 |
| BS position                    | (0, 0), (150, 150) |
| $E_{fs}$ (nJ/b)                | 50                 |
| $\varepsilon_{fs}$ (pJ/b/m²)  | 10                 |
| $\varepsilon_{mp}$ (pJ/b/m⁴)  | 0.0013             |
| $E_{da}$ (nJ/b/m²)            | 5                  |
| Data packet size (bits)        | 1000               |
| Initial energy (J)             | 2                  |
| Channel width                  | 250 kbps           |
| Second aggregation data numbers| 200 kbps           |

4. Experimental Analysis

4.1. Experimental Environment. Some algorithm parameter values are listed in Table 1. To determine the weight factor $a$, we have to analyze the number of rounds when the first node dies.

As shown in Figure 4, DADAT has a longest network lifetime when $a = 0.5$, so we take $a = 0.5$. In Section 4.3, we find the updating round numbers for PEDAP and PEDAP-PA algorithm which is 100, and it is too empirical to solve the problem. So we have to simulate for the optimal update numbers, and the results are shown in Figure 5, and the better updating round numbers is 90 which is adopted in our algorithm.

4.2. The Survival Node Numbers of the Network. In this section, we compare the algorithm performance among GIT, PEDAP, PEDAP-PA, and DADAT. Figure 6 shows the comparison of four algorithms with the survival node numbers in the same round. We find that DADAT can prolong the lifetime of the network, because DADAT in the establishment of a tree adopts the adaptive revision and the updating round number is optimized. Therefore, nodes in the same layer almost consume the same energy in each round and die almost in the same time, and no node is to be the bottleneck node to influence the transmission performance of the network. However, PEDAP and PEDAP-PA algorithms establish the tree under the condition of energy efficiency because some nodes in the tree die due to its high degree. After GIT builds the tree, it can be further optimized because there do not exist maintenance operations.

4.3. Network Delay. Figure 7 shows the network delay comparison among GIT, PEDAP, PEDAP-PA, and DADAT algorithms under the different networks scale. As shown in Figure 7, DADAT achieves a minimal delay because DADAT algorithm takes into account the number of hops to be the delay factor while the other three algorithms do not consider the distance factor and the node residual energy. However, the other three algorithms did not consider this factor of the number of hops, so it leads to the network average delay being too large.

4.4. The Energy Consumption. We compare the energy consumption of the four algorithms: GIT, PEDAP, PEDAP-PA, and DADAT. As shown in Figure 8, we can see that the trend of energy consumption is similar because they are all based on the minimum spanning tree topology. However, DADAT has minimum energy consumption than others; this because it is based on energy-aware simultaneously maintenance tree operations. The GIT only considers the distance factor, but the energy consumption and the updating tree operation are absent, so the tree has maximum energy consumption. The PEDAP-AP algorithm considers the node residual energy; therefore PEDAP-PA algorithm’s energy consumption is less than GIT and PEDAP, but PEDAP-PA does not frequently adopt the adaptive tree maintenance, so it is not as effective as DADAT.

In addition, we can conclude that our algorithm has good scalability from Figure 6 to Figure 8; also it displays some
good properties in different scenarios (sink node is located both in the center and on the edge).

5. Conclusion

In this paper, data collection problem is studied. We consider the number of hops, the residual energy, and aggregation factor, and then a distributed dynamic aggregation tree algorithm based on constrained-degree is established. The degree of nodes in the tree can be adjusted according to the average degree of the tree, and then all nodes are labeled by red, yellow, and green colors according to their remaining energy; red nodes can transfer their child nodes adaptively.

We finally discuss the weight and the updating rounds’ impact on the network lifetime. The simulation results show that our proposed approaches can effectively balance the energy consumption, prolong the network lifetime, and result in a lower latency.

In the next work we will consider heterogeneous data aggregation in wireless sensor networks, while meeting the QoS requirements of a boundary [13].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
FIGURE 8: The residual energy of different algorithm.

Acknowledgments

This project is supported by the National Natural Science Foundation of China (Grants nos. 60703118, 71271165, and 61373174), the Open Research Funds of CEMEE (Grants nos. CEMEE20120207B and CEMEE20140302A), and Zhejiang Province Research Funds (2010R50041 and 2011C14024).

References

[1] K. Kucuk, A. Kavak, H. Yigit, and C. Ozdemir, "A novel localization technique for wireless sensor networks using adaptive antenna arrays," in Proceedings of the IEEE Radio and Wireless Symposium, pp. 483–486, Orlando, Fla, USA, 2008.
[2] P. Kulakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," Computers and Electrical Engineering, vol. 36, no. 6, pp. 1181–1186, 2010.
[3] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," IEEE Wireless Communications, vol. 14, no. 2, pp. 70–87, 2007.
[4] P. Wan, S. C. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09), pp. 185–193, 2009.
[5] J. Zhang, Q. Wu, F. Ren, T. He, and C. Lin, "Effective data aggregation supported by dynamic routing in wireless sensor networks," in Proceedings of the IEEE International Conference on Communications (ICC '10), pp. 1–6, Cape Town, South Africa, 2010.
[6] M. Ding, X. Cheng, and G. Xue, "Aggregation tree construction in sensor networks," in Proceedings of the IEEE 58th Vehicular Technology Conference (VTC '03), vol. 4, pp. 2168–2172, Orlando, Fla, USA, 2003.
[7] H. Ö. Tan and I. Körpeoğlu, "Power efficient data gathering and aggregation in wireless sensor networks," SIGMOD Record, vol. 32, no. 4, pp. 66–71, 2003.
[8] W. Liang and Y. Liu, "Online data gathering for maximizing network lifetime in sensor networks," IEEE Transactions on Mobile Computing, vol. 6, no. 1, pp. 2–11, 2007.
[9] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm," in Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08), pp. 356–360, Washington, DC, USA, 2008.
[10] B. Krishnamachari, D. Estrin, and S. Wicher, "Modelling data-centric routing in wireless sensor networks," in Proceedings of the 21st IEEE Conference on Computer Communications (INFOCOM '02), pp. 1–11, 2002.
[11] X. Qi, Z. Zhang, and F. Han, "A novel clustering and routing algorithm for wireless sensor networks," International Journal of Advances in Computing Technology, vol. 3, no. 9, pp. 114–121, 2011.
[12] M. H. Yeganeh, H. Yousefi, N. Alinaghipour, and A. Movaghar, "RDAG: a structure-free real-time data aggregation protocol for wireless sensor networks," in Proceedings of the 17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '11), pp. 51–60, Toyama, Japan, 2011.
[13] D. Zhang, C.-P. Zhao, Y.-P. Liang, and Z.-J. Liu, "A new medium access control protocol based on perceived data reliability and spatial correlation in wireless sensor network," Computers and Electrical Engineering, vol. 38, no. 3, pp. 694–702, 2012.