An Energy-Efficient Self-Organization Routing Strategy in Tree Networks

LinFeng
Department of Computer Science,
Dalian University of Technology
Dalian 116024, China
+8613942650522
fenglin@dlut.edu.cn

YuZhou
Department of Computer Science,
Dalian University of Technology
Dalian 116024, China
+8613342288537
zhouyujoe@126.com

TieQiu
Department of Software,
Dalian University of Technology
Dalian 116620, China
+8613500705662
qiutie@ieee.org

ABSTRACT

This paper proposes a new self-organization and topology dynamic adjustment strategy. Network nodes broadcast packets to select child nodes and non-network nodes collect packets to join in network. During the self-organization process we take hop, residual energy, number of child node and communication distance into account to calculate the weight of available sink nodes, and then select the node with max weight as sink node. After a non-network node joins the network successful it will work as a network node to searching child nodes. A tree network can be constructed one layer by one layer using the new strategy. For balancing energy consumption and prolonging network lifetime we adjust the topology dynamic. All experiments were done with NS2.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Network Protocols – *routing protocols.*

General Terms

Algorithms, Performance, Design.

Keywords

Self-organization, Tree Network, Energy Balanced, Reorganization.

1. INTRODUCTION

WSNs (Wireless Sensor Networks) is a rapid self-organized distributed network, which is composed by a number of low-power and low-cost wireless sensor nodes [1].Tree network is a real-time network and can balance the energy consumption of nodes, so researchers construct maximum lifetime tree for WSNs. Zhu *et al.* in [2] have proved that we cannot create a tree within a polynomial time. Without data aggregation we can improve the real-time of network, but it increases the energy consumption and network load. In this paper we combine tree network with data aggregation technology to reduce the network load. At present, clustering algorithm is widely used. LEACH (Low Energy Adaptive Clustering Hierarchy) [3] and HEED (Hybrid Energy-Efficient Distributed Clustering) [4] are all based on clustering topology control. The cluster-header selection of LEACH and HEED are periodic. Chen *et al.* improves LEACH in [5]. However, the cluster-header selection only based on the residual energy of nodes and there is only one hop between cluster-header and BS (Base Station), which will increase the cluster-header reelection frequency. Afsar *et al.* in [6] proposes a new algorithm EEDC (Energy-Efficient Distance-based Clustering) to create cluster. The selection of cluster-header is based on the residual energy and distance of nodes. The network performance of EEDC is better than LEACH and HEED. ECBDA (Energy efficient Cluster Based Data Aggregation) [7] creates layer based on distance between nodes with BS and one cluster is one layer. New cluster-header is reelected only when old cluster-header needs to balance energy consumption. We reference this to reorganize for the hot area of the network.

In this paper we propose an Energy-efficient Self-organization Routing Strategy in tree network (ESRS). The root node is the first network node(the nodes that have joined in network) and its hop is zero. First, the root node broadcasts to searching child nodes and other non-network nodes(the nodes that have not joined in the network) deal with the received broadcast packets and record them. Then after a fixed period, non-network nodes calculate the weight of available sink nodes based on hop, residual energy, number of child node and distances, and then select the node with max weight as sink node to join the network. If a non-network node joins the network successful it will broadcast to searching child nodes. A tree network can be constructed quickly with this algorithm. During the working process of the network we adjust the topology dynamic to prolong the network lifetime. Removing the farthest child node and notifying it to reelect sink node, when we need to balance the energy consumption of the sink node.

The rest of this paper is organized as follows: Section 2 analyses the new algorithm ESRS. Section 3 is the implement of Section 2. The experiments and experimental results are discussed in Section 4. Section 5 is the conclusion of this paper.

MOBIMEDIA 2015, May 25-27, Chengdu, People's Republic of China

DOI 10.4108/icst.mobimedia.2015.258969

2. ENERGY-EFFICIENT SELF-ORGANIZATION ROUTING STRATEGY IN TREE NETWORK

2.1 Network Self-organization

Equation (1) of [8] is the energy model we use. In actual use the communication distance d is less than d_{cr}, so e and S are constant values. E_c is determined by communication device and we assume it is constant. If the length of data packet is fixed the energy consumption is determined by d. So selecting the closest sink node is benefit to energy saving. However, selecting the closest sink node may increase the average hop of network. With the increase of hop the data transmission will consume more energy and the real-time of the network will decrease. So we need take distance and hop of nodes into account to select sink node. The more child nodes the more received packet at the same time and it will consume more energy. So we need to balance the hop, residual energy, number of child nodes and distance of node to select the best sink node. Referring to (5), (6) of [9] and balancing these four factors we get (1) to calculate the weight of sink $node_i$.

$$W_i = \alpha / D_i + \beta / (N_i + 1) + \lambda E_i + \delta / (H_i + 1) \tag{1}$$

W_i is the weight of $node_i$ and it is the larger the better. D_i is the distance between current node with $node_i$ and it is the smaller the better. N_i is the number of child node of $node_i$ and it is the smaller the better. (N_i+1) is the degree of $node_i$. E_i is the residual energy of $node_i$ and it is the greater the better. H_i is the hop of $node_i$ and it is the smaller the better. (H_i+1) deals with the root node whose hop is 0. α, β, λ, δ are normalized parameters of these four factors.

2.2 Adjust Topology Dynamic

The energy consumption of sink node is quicker than sensor nodes and the more child nodes the quicker. After some time of data transmission, if the energy of a sink node reduces to R% we need to delete the farthest child node to balance the energy consumption. Energy percentage R is related to the number of child nodes number and the calculation equation as (2). N is the number of child node. Why we delete the farthest child node: firstly, according to (1) of [8] the farther the more communication energy. Secondly, the farther the smaller probability of reelect this node as sink node.

$$R = N / (N + 1) \tag{2}$$

3. ALGORITHM DESIGN

All packet types of this paper are shown in Table 1.

Table 1. Packet type definetion

Symbol	Description
PT_JOIN_REQUEST	Packet of non-network node requests to join in network.
PT_ACCEPTED	Packet from sink node to accept the non-network node joins in network.
PT_DENIED	Packet from sink node to deny the non-network node joins in network.
PT_JOIN_OK	Reply of PT_ACCEPTED.
PT_DELETE	Packet from sink node to delete a child node.
PT_DELETE_OK	Reply of PT_DELETE.
PT_SINK_SEARCH	Broadcast packet of non-network node to search available sink node.

3.1 Network Self-organization

During the process of network self-organization, nodes are divided into two kinds: network node and non-network node.

Network Node:

In the network self-organization phase, network node sends some broadcast packets to searching child node actively. After that the node will switch into network monitor state. During this process the node needs to check whether it receives PT_JOIN_REQUEST packet from non-network node and whether the array child[](An array to record information of current child nodes) has available space. If it receives PT_JOIN_REQUEST packets from a non-network node and the array child[] has available space it will send PT_ACCEPTED packet to the non-network node. Otherwise it will send PT_DENIED packet to the non-network node.

Non-network Node:

All non-network nodes are in sleep model before they receive the first packet from network node. If a non-network node receives a packet from network node for searching child node, it will start a timer and saves the information of available sink node in array optional_sink[](An array to record the information of available sink nodes). When the timer is over the non-network node scans array optional_sink[] and selects the best sink node based on (1). Then it sends PT_JOIN_REQUEST packet to the best sink node and waits the reply. If it gets a reply of PT_ACCEPTED it needs to send a PT_JOIN_OK packet to the best sink node. Here this non-network node joins in the network succeed and becomes a network node, it will broadcast packet to search child nodes. If the best sink node replies a PT_DENIED packet that means this non-network node cannot select this best sink node as its sink node. The non-network node needs to reelect sink node from other available sink nodes until it receives PT_ACCEPTED packet. After scanning all available sink nodes and the non-network node still cannot join in the network, it needs to clear array optional_sink[] and resets timer to wait for searching of other network nodes. All non-network nodes can select a best sink node rapidly. After that they will send packets to request join in network. The detail information is shown in Algorithm 1.

Algorithm 1. Non-network node requests join in network

```
Begin
    Step 1. Send PT_JOIN_REQUEST packet to node optional_sink[sink_index] and start a timer.
    Step 2. The node receives and saves packets before the timer is over.
            Step 2.1. If the node received PT_ACCEPTED packet from its available sink node it needs to send PT_JOIN_OK packet to reply, and then go to Step 4.
            Step 2.2. If the node received PT_DENIED packet from its available sink node it needs to reelect a sub-optimal sink node.
    Step 3. Reelect a sub-optimal sink node, i=0, max_weight=0, sink_index_tmp=0(A variable to record the index of sink node in array optional_sink[]).
            Step 3.1. If i≥ava_sink_num go to Step 3.6.
            Step 3.2. If optional_sink[i].weight≤optional_sink[sink_index].weight && i!=sink_index, go on; else go to Step 3.5.
            Step 3.3. If optional_sink[i].weight>max_weight, then do sink_index_tmp=i, max_weight=optional_sink[i].weight, and then go to Step 3.5.
            Step 3.4. If optional_sink[i].weight==max_weight,
```

separate compare the hop, ratio value of residual energy with number of child node, distance of optional_sink[i] and optional_sink[sink_index_temp], sequentially. The priority selection standard is less hops, greater ratio of ((residual energy)/(number of child node+1)), less distances. If the node of optional_sink[i] is better, do sink_index_temp=i, max_weight= optional_sink[i].weight.

Step 3.5. i++, go to Step 3.1.

Step 3.6. If max_weight==0 it means there is no available sink node and the non-network node cannot join the network, then do ava_sink_num=0 and go to End to wait for other nodes' searching; else do sink_index=sink_index_tmp, go to Step 1 to rejoin network.

Step 4. Non-network node joins in the network successful and becomes a network node.

End

There is a loop in Step 3 to scanning array optional_sink[] and the array length is ava_sink_num. ava_sink_num is numerable and the complexity of Algorithm 1 is O(n).

3.2 Reorganization of Hot Area

When a sink node adds or deletes a child node it will update E_{org}(The residual energy of last topology change) with E_{ava}(Current residual energy). For prolonging the network lifetime and balancing the energy consumption all sink nodes need to check their energy consumption and Algorithm 2 is the check method.

Algorithm 2. Check energy consumption of sink node

Begin

Step 1. Check whether $(E_{ava}/E_{org}) \leq (R=N/(N+1))$ when the energy check timer is over, if $(E_{ava}/E_{org}) \leq (R=N/(N+1))$ select the farthest child node and send PT_DELETE packet to it. At the same time update E_{org}: $E_{org}=E_{ava}$. Otherwise go to End.

Step 2. After sink node sends PT_DELETE packet it needs to wait for the PT_DELETE_OK packet. If the node receives PT_DELETE_OK packet it will delete the farthest child node's record and update N: N=N-1.

Step 3. If N==0 the sink node becomes a sensor node, else start a new energy check timer.

End

There is no loop in Algorithm 2 so the complexity is O(1), but after sending a PT_DELETE packet the node starts a timer to wait the reply and it will cost some time. If a node receives a PT_DELETE packet or doesn't receive a reply packet during some time it needs to reelect sink node. Firstly, it broadcasts some PT_SINK_SEARCH packet periodic and start a timer. During the timer, if the node receives reply of PT_SINK_SEARCH packet it will save the sink node's information in array optional_sink[] and update ava_sink_num. Secondly, after the timer it scans the array optional_sink[] to select the best sink node and then join in the network based on Algorithm 1. If the node's hop is changed it needs to inform its child nodes to update their hop.

4. SIMULATION AND ANALYSIS

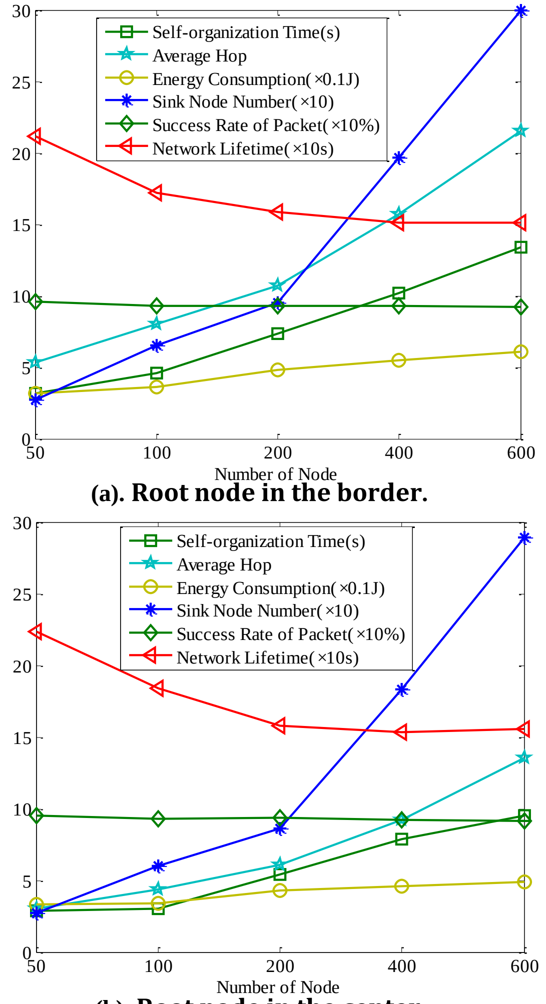
In this paper all experiments are all based on NS2. Following experiments are all based on ESRS protocol. According to the experiment settings, we set $\alpha = 15$, $\beta = 11$, $\lambda = 1/29$ and $\delta = 11$.

In order to verify the efficiency of ESRS we do five groups experiments. All nodes are randomly distributed in the test area and the number of nodes is 50, 100, 200, 400 and 600, respectively. The size of the test area and root node site sees Table 2.

Table 2. Experiment Settings

Number of Node	Test Area Size (m×m)	Root Node in Border (m,m)	Root Node in Center (m,m)
50	70×70	(0,0)	(35,35)
100	100×100	(0,0)	(50,50)
200	141×141	(0,0)	(70.5,70.5)
400	200×200	(0,0)	(100,100)
600	245×245	(0,0)	(122.5,122.5)

Figure 1 is the experiment results and experiment results show that sink nodes are about 50% of all nodes. So the percent of important nodes is less than 50% and we set P=50 is feasible. Packet loss is inevitable so the packet loss ratio is a very important indicator of network performance. With the network size increasing the success rate of packet doesn't decrease significantly and it's more than 90%. So ESRS is reliable. The network performance is worse when the root node in the border. With the node increasing the network lifetime has no great changes.



(a). Root node in the border.

(b). Root node in the center.

Figure 1. Experiment results of ESRS.

5. CONCLUSION

This paper proposes ESRS to balancing the energy consumption and prolonging the network lifetime, and at the same time building a reliable tree network. During the self-organization process we take hop, residual energy, number of child node and distance into account to calculate the weight of available sink nodes, and then select the node with max weight as sink node. After the network is constructed all nodes begin to collect and transmit data. The energy consumption of sink node is faster and for prolonging the lifetime of network we reorganize for the hot area of network. From the NS2 simulate experiments results we know that ESRS can build a reliable tree network quickly.

6. ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundations of China Nos. 61173163, 61202443, 51105052 and 61370200.

7. REFERENCES

[1] Olascuaga-Cabrera, J., Lopez-Mellado, E., Mendez-Vazquez, A., & Ramos-Corchado, F. F. 2011. A Self-Organization Algorithm for Robust Networking of Wireless Devices. *Sensor Journal. IEEE.* 11, 3, 771-780.

[2] Zhu, X., Wu, X., & Chen, G. 2015. An exact algorithm for maximum lifetime data gathering tree without aggregation in wireless sensor networks. *Wireless Networks*. 21, 1, 281-295.

[3] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. 2002. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications. IEEE Transactions on.* 1, 4, 660-670.

[4] Younis, O., Fahmy, S. 2004. Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies* (Vol. 1). (Mar. 2004), 629-640. IEEE.

[5] Chen, Y. L., Wang, N. C., Shih, Y. N., & Lin, J. S. 2014. Improving Low-Energy Adaptive Clustering Hierarchy Architectures with Sleep Mode for Wireless Sensor Networks. *Wireless Personal Communications,* 75, 1, 349-368.

[6] Afsar, M. M., & Tayarani-N, M. H. 2014. A Novel Energy-Efficient and Distance-Based Clustering Approach for Wireless Sensor Networks. In *Soft Computing in Industrial Applications,* 177-186. Springer International Publishing.

[7] Ranjani, S. S., Krishnan, S. R., Thangaraj, C., & Devi, K. V. 2013. Achieving Energy Conservation by Cluster Based Data Aggregation in Wireless Sensor Networks. *Wireless personal communications,* 73, 3, 731-751.

[8] Khamforoosh, K. 2011. Clustered Balanced Minimum Spanning Tree for Routing and Energy Reduction in Wireless Sensor Networks. In *Wireless Technology and Applications (ISWTA), 2011 IEEE Symposium on* (Sep. 2011), 56-59. IEEE.

[9] Zhang, H., Chen, P., & Gong, S. 2010. Weighted Spanning Tree clustering routing algorithm based on LEACH. In *Future Computer and Communication (ICFCC), 2010 2nd International Conference on* (Vol. 2). (May 2010), 223-227. IEEE.