A Weight Cluster-Based Hybrid Routing Algorithm of ZigBee Network

Yan Li, An-na Yuan, Xue Liu, Yong-bin Du, Tian-xin Huang and Hao-xin Cui

The Computer and Control Institute, Harbin University of Science and Technology, China
liyan@hrbust.edu.cn

Abstract

Aiming at the premature paralysis of network problem, which is caused by energy consumption imbalance of nodes in ZigBee, a weight-based clustering hybrid routing algorithm is proposed. From the viewpoint of energy, the algorithm considers the status of the node and network, selects cluster head according to the weights, and achieves energy balanced clustering of heterogeneous nodes. In the cluster, we adopt cluster-Tree routing algorithm to implement optimization of local network. And among the different clusters, we adopt the AODVjr algorithm to reduce the latency of communication between the different clusters and routing overhead by controlling the flooding direction of RREQ. The simulations reveal that the improved algorithm effectively reduces the mortality rate of nodes and decreases energy consumption in the network, balances the network energy load, and prolongs the lifetime of network.

Keywords: ZigBee network; Cluster-Tree algorithm; AODVjr algorithm; clustering; energy balance

1. Introduction

ZigBee technology is a kind of wireless communication technology based on IEEE802.15.4 standard. Mainly used in short distance, the technology has low power consumption and low transmission rate of message transmission and communication between electronic devices. Therefore, energy saving becomes one of the main goals of ZigBee network routing protocol design. This paper proposed a hybrid routing algorithm based on weighted clustering for load-balancing network energy and reducing energy consumption, to prolong the purpose of the network lifetime.

2. Analysis of ZigBee Routing Algorithm

2.1. Cluster-Tree Algorithm

Cluster-Tree routing algorithm is the static routing algorithm which packet is forwarded base on hierarchical tree structure. When a node successfully joins the network, the parent will assign the unique address to it. For any full function device (FFD) with the depth d and the address A, if the destination node is descendant of the current device, then it is forwarded to the child node, otherwise, it is forwarded to the parent node. So the descendant node and the hierarchical route algorism can be assured as follows:

$$A < D < A + Cskip(d - 1) \tag{1}$$

If equation is not satisfied, the address of next hop is the address of its parent node. Otherwise, it realizing that the destination is the descendant of current node, then the address of next hop is calculated from the following equation:

$$N = \begin{cases} D & \text{the destination is a child of recent node} \\ A + 1 + \left\lceil \dfrac{D - (A + 1)}{Cskip(d)} \right\rceil \times Cskip(d) & \text{otherwise} \end{cases} \tag{2}$$

2.2. AODVjr Algorithm

AODVjr algorithm is simplified improved version of AODV (Ad-hoc On-demand Distance Vector) protocol. On the basis of succeeding main functions of AODV, it threw off sequence number and the seasonal HELLO packet of AODV in order to relieve the pressure of network bandwidth, and further promotes the aspects such as cost, energy conservation, and application convenience.

3. Description of the Problem

ZigBee uses both Cluster-Tree and AODVjr routing algorithms, but it does not provide an efficient mechanism to integrate these two algorithms, so they work comparatively separate. Ironically, if one uses these two algorithms at the same time, it cannot improve the efficiency of routing process on the contrary it put a burden on the whole network. In Cluster-Tree routing algorithm, if a child node wants to send data, the data should be sent by its parent node. Therefore, FFDs in the upper level of a network is easy to exhaust their energy and become dead nodes by frequent and overweight communication. If too many upper level nodes died, the network will easily be separated to several unconnected sub-network and lifetime of the network will be shorten dramatically. On the other hand, in AODVjr routing algorithm, although it can easily find the best routing path, all nodes are involved in routing discovering process to forward route request (RREQ) packet. This process generates large amount of redundant RREQ messages will waste resource of the network and cause network congestion. Take situation shown in Figure 1 as an example. The hub of the network is central coordinator and the network is separated into four clusters, where four cluster heads are marked as 0, 1, 2, 3 and cluster heads designate them which member nodes need to forward data messages.

Suppose the node 5 in the cluster1 as the source node wants to transmit data with destination node 6 in its adjacent cluster3. From the viewpoint of cost, the cost of transmission between neighbor nodes is 1. However, if based on Cluster-Tree routing algorithm, the data must be forwarded by their parent nodes, which means the total cost will be 5. From the viewpoint of energy-efficient, the former method only cost energy of node 5 and node 6 themselves, but the latter cost extra energy of node 4, node 3, node 0, node 1. What's worse, it is easily to deplete the energy of node 1 and node 4, as they are always involved in transmission, and eventually cause the separation of the network. Although it is the routing path of minimum cost based on AODVjr, it has its drawback on energy efficiency. When in the route discovering process, only the RREQ packet between node 5 and node 6 of all the RREQ packets across the network are useful, the rest of them are redundant packets which are the main reasons of network congestion.
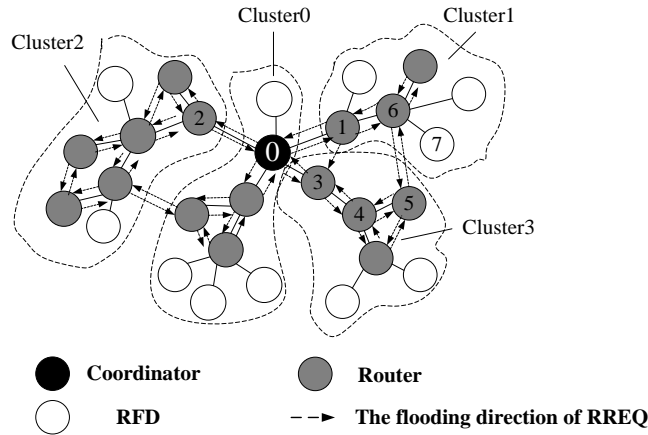
Figure 1. Topography of ZigBee Network

4. Design of Improved Algorithm

Considering the huge diversity of energy consumption of nodes, it is hard to balance the energy consumption of every FFD based on the original algorithm. In the improved algorithm, a new mechanism is raised by introducing the concept of remaining energy, energy ratio and weights. The algorithm selects node which has less weights as cluster head, preventing FFD with small amount energy from forwarding too much message as well as balancing loads and energy consumption of the whole network. This algorithm also optimizes the AODVjr routing algorithm to shorten routing distance and total energy consumption and maintain the lifetime of the entire network.

4.1. Definitions of Remaining Energy and Energy Ratio

Suppose the initial energy of a FFD node V_i is E_0, the depth of node V_i in the network is d_i, the formula of remaining energy $E(V_i)$ is

$$E(V_i) = E_0 - \frac{kt}{(d_0 + 1)} \tag{3}$$

Where, $\frac{kt}{(d_i + 1)}$ is the energy consumption in time t; d_i is the depth of node V_i; k is the parameter of controlling the decrease speed of energy and k of cluster head is larger than other router nodes. The remaining energy of the node continues decreasing with the passage of time t, and the less deep the node, the less small amount of remaining energy.

The definition of energy ratio is

$$P(V_i) = \frac{E(V_i)}{E} \tag{4}$$

Energy ratio $P(V_i)$ is the ratio of the remaining energy of FFD node $E(V_i)$ to the average remaining energy of the entire network, which reflects the energy state of the node in the network. The node with larger value of $P(V_i)$ represents it has more energy

than other nodes, which means it is more suitable for complex work such as forwarding large amount of message or more chance to be the next cluster head. The formula of average remaining energy is

$$\bar{E} = \frac{1}{N} \mathring{a} E(V_i) \tag{5}$$

4.2. Definition of Energy Metrics

Formula of metrics of nodes is

$$W(V_i) = a_1 \cdot \frac{1}{N(V_i)} + a_2 \cdot \log\left[\frac{(T(V_i) + 0.01)}{T_{thr} + 0.01}\right] + a_3 \cdot \frac{1}{P(V_i)} \tag{6}$$

Where $N(V_i)$ the number of neighbor node that one node has and the more neighbor one has, the less delay time and cost of forwarding in the cluster; $T(V_i)$ is the time that this node is the cluster head; T_{thr} is threshold time, which is when the time that a node has been cluster head is greater than T_{thr}, the its chance to be cluster head decrease dramatically and here it decrease in a way of logarithm (T_{thr} plus 0.01 is for avoiding logarithm equals ; $P(V_i)$ is the energy ratio; a_1, a_2, a_3 are parameters and satisfy equation $a_1 + a_2 + a_3 = 1$.

4.3. Implementation of Improved Routing Algorithm

Suppose a network is established according to the rules of ZigBee, the connection of nodes and children and parent relations has been determined and each node has been allocated IP address from its parent node. Then the network enters to clustering phrase.

1. Initialize the table of cluster head S_{CH} and list of members in the cluster $M(CH_i)$; central coordinator begins to advertise message of cluster head competing to the network and wait for nodes to apply for cluster head.

2. After receiving the message sent by central coordinator, every single node judge itself is function. If it is a Reduced Function Device (RFD), it will add itself to the cluster its parent node belonged to. If it is a FFD node, then go to next step.

3. Read the energy depth d_i, and judge whether its energy ratio is greater than 0.8 or not according to formula (4) in order to keep enough energy to apply for cluster head. If its energy ratio is less than 0.8, it will add itself to the cluster its parent node belonged to. Otherwise, go to next step.

4. Every single node advertises request message RREQ, and then other nodes will receive the message and reply acknowledge message RREP (Route Reply). The received node will add requesting node to its neighbor table and send it to central coordinator, and apply for cluster head.

5. Central coordinator waits for a while and then calculates the metrics of each applying nodes based on records of cluster head application and formula. The node with the least value of metrics will be the new cluster head; the rest of them will be added to the cluster its parent node belonged to.

6. New cluster heads sends broadcast messages; other nodes that are not cluster head receive the messages and reply add messages back to its own cluster head. After cluster heads receive these responses, they will add replying nodes to $M(CH_i)$ and exchange routing information with member nodes and member nodes need to save information about and maintain connection with its cluster head.

7. Clustering ends.

After clustering ends, the central coordinator sends all $M(CH_i)$ tables to each cluster heads, so they can have the topology of the whole network and information about each cluster heads, which enters transmission phase. Transmission phase is mainly to determine and establish the shortest route from the source node to destination node. Routing algorithm in transmission phase is as follows.

1. Judge a node whether a RFD node or a FFD node. If it is a RFD node, then send data to its parent node and depend on its parent node to forward the data. If it is a FFD node, then look up whether there is a node that can connect to destination node in its routing table. If such routes exist, then forward data directly, otherwise go to next step.

2. Look up routes to the cluster head of destination node. If such routes exist, forward data to this cluster head, otherwise look up whether forward to its cluster head by send the data to its parent node.

3. The cluster head looks up the route to destination node in its routing table. If a route exists, forward the data immediately, otherwise, look up whether there is a route to the cluster head of the destination node.

4. If there is one, forward the data to the cluster head. If not, broadcast routing information request messages (RREQ) to adjacent nodes, and enter neighbor discover phase.

5. At the start of neighbor discover phase, when a node receive the RREQ packet, first it should decide whether itself is the destination node or the parent node of destination node. If it is, add the information of the cluster head of the source node, and reply packet (RREP), otherwise go to next step.

6. Look up whether there is a route to the cluster head of the source node. If there is, compare the cost of the route to the cluster head in RREQ packet with that in its routing table; If there is not, create a record of RREQ, and then compare the cost. If the cost in RREQ packet is higher, discard this packet and routing information, in order to avoid forwarding redundant routing information and consuming energy in the network. Otherwise, replace the original routing information, update routing table and broadcast RREQ packet until it reach to destination node.

7. All the nodes that this updated routing information need to update their own routing table. When receiving this RREP packet, the source node creates and returns its cluster head a route notified message (RNOT) containing the routing information. When the cluster head receives this, it will broadcast this as a route update message (RURT) to member nodes in this cluster. Member nodes receive these messages and update the new routing information.
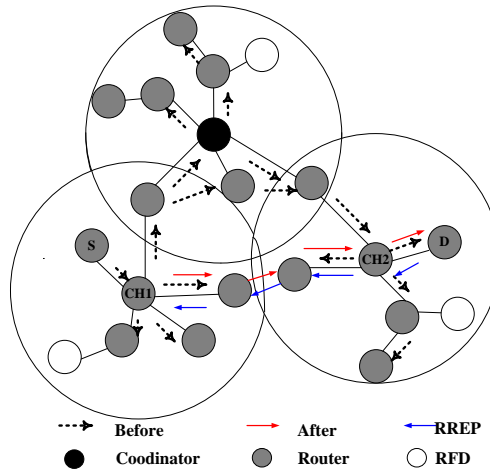
8. Transmission routing process ends.

Figure 2 Comparison of Routing Paths of RREG Messages

Suppose node S need to transmit data with node D, which shows in **Figure 2**. If only based on AODVjr algorithm, after receiving a RREQ packet, all nodes that do not have the route to destination node will broadcast RREQ packet to its adjacent nodes, which will create a lot of redundant RREQ packet. However, after adopting the improved algorithm, only cluster head broadcast RREQ packet and other nodes calculate and create the shortest route to the cluster head of the destination node, so indirectly create a route to destination node. In this way, redundant RREQ packet, cost of the whole network, chances of collusion in the network will be reduced, so it saves energy and avoids broadcast storm.

5. Simulation and Performance Analysis of the Algorithm

This paper simulates and compares the performance of improved Cluster-Tree algorithm and the original one in NS2. Simulating environment has been set up as: the range of the network is 100m*100m, the distance of between two nodes is 10m, the number of total nodes is 200, parameters of metrics are a_1=0.25, a_2=0.25, a_3=0.5 initial energy of each node is 50J, CBR bit is the data source and messages of data are 70 byte. The network has been set up as: L_m the depth of the network is 6, C_m the maximum number of nodes that a parent node can have is 5, and R_m the maximum number of a routing node that a parent node can have is 4.
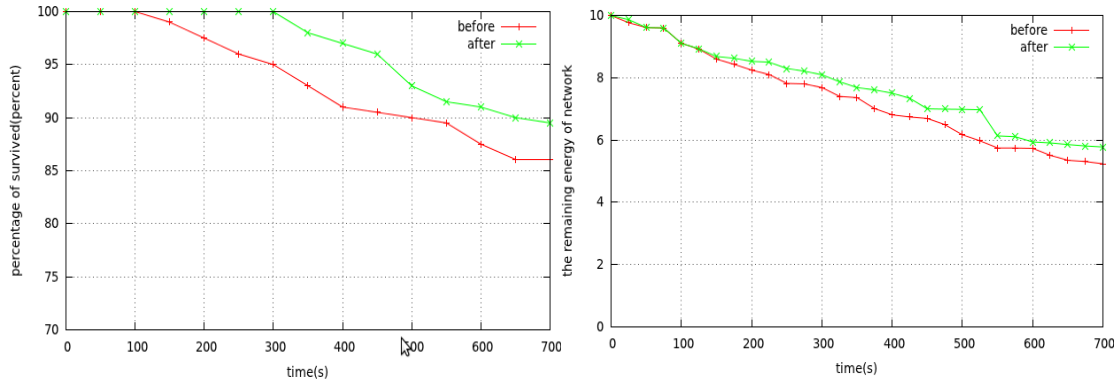
Figure 3. Comparison of Percentage of Survived Nodes and Remaining Energy

In the simulating experiment, a node is dead when its energy is below 5% of initial energy. One can see in the left graph of **Figure 3** that the percentage of survived nodes decreases with time passed by. The improved algorithm tries to avoid those nodes with small amount of energy as intermediate nodes transmit data in order to decrease the chance of its death. So it delays the time when dead nodes occur for 200s, and at any time the number of dead node in improved algorithm is less than that in original one, which achieve balance the energy consumption.

The right graphic of **Figure 3** is the comparison of total remaining energy before and after adopting improved algorithm. Total energy consumption contains energy for clustering, forwarding RREQ packet and forwarding data. The new algorithm limits flooding, reduces the number of hops between the source node to the destination node and creation of redundant messages and energy consumption. Thus, improved algorithm increases the remaining energy of network about 3.6%.
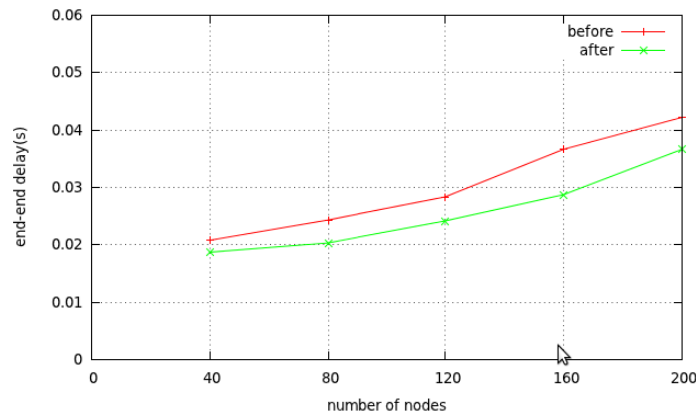

Figure 4. Comparison of end-to-end Delay

Figure 4 shows the linear function of end-to-end delay to the number of nodes, which is with the increase of the number of nodes, the end-to-end delay increase as well. Due to the optimization of routing form the source node to the destination node, the average distance of data message has been shorten, improving the situation of network congestion and delaying.

6. Conclusion

This paper comprehensively analyzes the routing algorithm in ZigBee protocol. Based on the its drawback, an improved algorithm combining the new-created clustering mechanism and optimized AODVjr routing algorithm is raised. This paper also simulates and compares the performance of the improved algorithm with the original one in NS2. According to the result of NS2, this new improved algorithm can increase efficiency of the whole network, balance the cost of each node, optimize the performance of data transmission and elongate the life cycle of the network.

Acknowledgements

References

[1]   K. K. Lee, S. H. Kim and H. S. Park, "Cluster Laber-based ZigBee Routing Protocol with High Scalability", Proceedings of the Second International Conference on Systems and Networks Communications (ICSNC 2007), Cap Esterel, France, **(2007)** August 25-31.

[2]   K. K. Lee, S. H. Kim, Y. S. Choi and H. S. Park, "A Mesh Routing Protocol using Cluster Label in the ZigBee Network", Proceedings of 2006 IEEE internation Conference on Moblie Adhoc and Sensor Systems (MASS), Vancouver, BC, **(2006)** October.

[3]   Q. Y. Pan, J. Wu, Y. H. Wang and J. F. Ni, Journal of Software Engineering and Applications, vol. 4, no. 487, **(2011)**.

[4]   A. Khatiri, G. Mirjalily and A. Khademzadeh, "Energy-Efficient Shortcut Tree Routing in ZigBee Networks.Computational Intelligence", Proceedings of 2012 Fourth International Conference on IEEE Communication Systems and Networks (CICSyN), Phuket, Thailand, **(2012)** July 24-26.

[5]   H. Abusaimeh and S. H. Yang, International journal of automation and computing, vol. 6, no. 48, **(2009)**.

[6]   Z. H. Qian, S. Zhu and X. Wang, Chinese Journal of computers, vol. 3, no. 485, **(2013)**.

[7]   W. F. Li, L. F. Shen and J. Hu, Journal on Communications, vol. 33, no. 10, **(2012)**.

[8]   X. Y. Liu, B. Li, S. Huang and M. Chen, Computer Engineering, vol. 38, no. 97, (2012).

[9]   Y. Wang, Z. H. Qian, L. H. Feng and Y. Q. Guo, Advanced Materials Research, vol. 765, no. 1667, **(2013)**.

[10]   Y. Yao, Z. Zhang, Z. Ren and S. Shi, "Neighbor-Information-Assisted Tree Routing for ZigBee Networks", Proceedings of 2006 IEEE internation Conference on Computer Science and Electronics Engineering (ICCSEE), HangZhou, China, **(2012)** March 23-25.