A Borrowed Address Assignment Algorithm Based on the Depth Model for ZigBee Network

[1]Leqiang BAI & [2]Zhenhu ZHANG

Information & Control Engineering Faculty
Shenyang Jianzhu University, Shenyang, China
[1]*baileqiang@sjzu.edu.cn,* [2]*zhangzhmars@163.com*

ABSTRACT: Aiming at the problems of the shortcomings of the ZigBee distributed address assignment mechanism lead to orphan nodes and the lower success rate of address assignment, a borrowed address assignment algorithm based on the depth model for ZigBee network is proposed. The algorithm utilizes 16-bit network address that DAAM algorithm unused to assign addresses for different network depth orphan nodes. Meanwhile, the tree routing algorithm to suit the address assignment mechanism is proposed. Theoretical analysis and simulation results show that, in the condition of the same computational complexity and the same maximum network depth, the algorithm outperforms DAAM, SLAR, DAAM-THN and SOSSA in terms of the success rate of address assignment, and the average time of assigning address etc.
KEYWORD: ZigBee network; Orphan node; Address assignment; Depth model; Tree routing

1 INTRODUCTION

ZigBee protocol is one of the Wireless sensor network protocol based on IEEE802.15.4 wireless protocol [1]. When the routing child numbers and end child numbers of a router are greater than R_m and C_m-R_m respectively, some nodes cannot obtain the address, which known as "orphan nodes" [2].

Recently, many improved algorithms based on borrowed address mechanism have been proposed. Ref [3], which enlarges the address sub-block by altering the parameter $C_{skip}(d+1)$ instead of $C_{skip}(d)$. $C_{skip}(d)$ is defined in formula (1).

$$C_{skip}(d) = \begin{cases} 1 + C_m \times (L_m - d - 1) & , R_m = 1 \\ \dfrac{1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 - R_m} & , R_m \neq 1 \end{cases} \quad (1)$$

As in Ref [4], when a router lacks addresses, it extended its address space to a new segment. The address A_n of its mth child node is:

$$A_n = \begin{cases} A_s \times A_{DAAM} + A_{parent} + C_{skip}(d) \times m + 1, & end\ devices \\ A_s \times A_{DAAM} + A_{parent} + C_{skip}(d) \times R_m + m + 1, & routing\ devices \end{cases} \quad (2)$$

Where A_s = the number of address extended operation; A_{DAAM} = the maximum address number which DAAM assigned for the network; A_{parent} = the parent address; and m = the number of the same type nodes in the same segment address space. Formula (2) shows that there is a phenomenon of wasting address when the routing node with address A_{parent} assigns address for orphan nodes using offset $C_{skip}(d)$. As in Ref [5], which making orphan node send address request message to the routing node in the two-hop range, so the algorithm increases the additional communication overhead. As in Ref [6], orphan node borrows address firstly from the offspring nodes in the same branch, it decreases the communication overhead.

In order to solve the problems of the shortcomings of the ZigBee distributed address assignment mechanism, a borrowed address assignment algorithm based on the depth model for ZigBee network is proposed (BAADM).

2 DISTRIBUTED ADDRESS ALLOCATION MECHANISM ALGORITHM AND TREE ROUTING ALGORITHM

2.1 *Distributed address allocation mechanism*

The ZigBee network uses a distributed address assignment mechanism to assign a unique network address for each node, its steps are as follows:
Coordinator establishes a network according to the three network parameters which are C_m, R_m and L_m.

Step 1. Coordinator or router (A_{parent}) broadcasts networking message.

If the network depth of A_{parent} not reached the value of L_m or the number of the routing child nodes and the end child nodes not reached R_m and C_m-R_m respectively, A_{parent} broadcasts networking message. Otherwise, A_{parent} does not do that.

Step 2. Address requesting of the no address nodes.

Step 2.1. No address node sends address request message for joining the network.

After a no address node has received the networking message, it sends address request message to the potential parent node through inquiring the neighbor table to find the unasked potential parent node which has the least network depth (select a node which has a larger RSSI if there are several).

Step 2.2. No address node receives a response message from the potential parent node.

If the no address node receives address from the potential parent node, and the end. If it receives a no idle address message and carries out Step 2.1, until it traverses the potential parent nodes in the 1-hop neighbor range.

Step 3. Coordinator or router replies address request message.

The potential parent node inquires its address space after having received address request message. If the potential parent node has no free address, it replies no idle address message. Otherwise, it implements DAAM and assigns address for the different types of child node according to formula (3) and (4).

The address A_n of its nth router node is:

$$A_n = A_{parent} + C_{skip}(d) \times (n-1) + 1, \ 1 \leq n \leq R_m \quad (3)$$

The address A_n of its nth end node is:

$$A_n = A_{parent} + C_{skip}(d) \times R_m + n, \ 1 \leq n \leq C_m - R_m \quad (4)$$

2.2 Tree routing algorithm

In the tree routing algorithm, when the source node A_n forwards packets to the destination node D, the tree routing algorithm firstly determines whether the destination node is the descendant node of A_n. If the destination node is the node A_n's descendant node, the packets are forwarded to the corresponding child node. The necessary and sufficient condition of determining whether the destination node D is the descendant node of node A_n, it is given by:

$$A_n < D < A_n + C_{skip}(d-1) \quad (5)$$

Under the premise of formula (5), the next hop node address N of the node A_n as follows formula (6):

$$N = \begin{cases} D, & \text{if the destination node is the child node of } A_n \\ A_n + 1 + \left\lfloor \dfrac{D - (A_n + 1)}{C_{skip}(d)} \right\rfloor \times C_{skip}(d), & \text{others} \end{cases} \quad (6)$$

If the destination node is not the descendant node of A_n, according to tree structure, packets will be forwarded to node A_n's parent node.

3 BAADM ALGORITHM DESIGNING

According to the network properties which the network depth increase makes the lower average address utilization ratio of every node [7], the BAADM algorithm was proposed. The parameters of BAADM algorithm are defined as follows:

Definition 1. Let $S_k = \{s_i\}$ be a set of no address nodes after using DAAM algorithm, $1 \leq i \leq 2^{16} - A_{DAAM}$.

Definition 2. Let β represent the address numbers which router reserved for node s_i from the unused remainder address block ($2^{16} - A_{DAAM}$) by DAAM in the 16 bit address space, it is given as:

$$\beta = \left\lfloor \frac{2^{16} - A_{DAAM}}{\sum_{d=0}^{L_m-1} R_m^{L_m-d-1}} \right\rfloor \quad (7)$$

Definition 3. Let $C_{offset}(d)$ represent the borrowing address offset which the parent node at depth d ($d \leq L_m - 1$) assigns for its branch, it is given as:

$$C_{offset}(d) = \beta \times \sum_{j=d}^{L_m-1} R_m^{L_m-j-1} \quad (8)$$

Definition 4. Let n_d represent that node s_i's parent node (depth is d, address is A_{parent}) is the n_d^{th} child node of A_{parent}'s parent node (depth is $d-1$, address is A_{pp}) [7], it is given as:

$$n_d = \begin{cases} \left\lfloor \dfrac{A_{parent} - (A_{pp} + 1)}{C_{skip}(d-1)} \right\rfloor, & 1 \leq d \leq L_m - 1 \\ 0, & d = 0 \end{cases} \quad (9)$$

Definition 5. Let A_n represent the potential parent node at depth d assigns address for the mth child node using BAADM algorithm, it is given by:

$$A_n = A_{DAAM} + \sum_{j=0}^{d} n_j \times C_{offset}(j) + d \times \beta + m, \ 1 \leq m \leq \beta \quad (10)$$

3.1 BAADM algorithm

The BAADM algorithm's steps are as follows:

The 1, 2 and 3 steps of BAADM and DAAM algorithm are the same.

Step 4. Borrowed address allocation mechanism. Node s_i sends address request message to the potential parent node, the potential parent node assigns address for node s_i from the unused remainder address block ($2^{16} - A_{DAAM}$) by DAAM in the 16 bit address space.

Step 4.1. Address request process of node s_i.

Step 4.1.1. Node s_i sends address request message for joining the network.

After node s_i has received the networking message, it sends address request message to the potential parent node through inquiring the neighbor table to find the unasked potential parent node which has the least network depth.

Step 4.1.2. Node s_i receives response message from the potential parent node.

If node s_i receives address assignment message from the potential parent node, then the end. If nodes s_i receives no free address request message, and carries out step 4.1.1, until a no address node traverses the potential parent nodes in the 1-hop neighbor range.

Step 4.2. Potential parent node A_{parent} replies address request message.

The potential parent node A_{parent} after receiving a address request message from node s_i determines Whether the network depth of A_{parent} has reached the value of L_m, if it has reached L_m, it replies the no free address message; Otherwise, the potential parent node inquires its free address space, if it has no free address, it replies the no free address message. If it has free address, it assigns address for node s_i according to formula (10). Meanwhile, the free address number subtracts 1.

3.2 *Tree routing algorithm based on BAADM algorithm*

The steps of the tree routing algorithm based on BAADM algorithm are as follows:

Step 1. If $A_n \leq A_{DAAM}$ and $D \leq A_{DAAM}$, it forwards packets using the tree routing algorithm based on DAAM; Otherwise, go to step 2.

Step 2. If $A_n \geq A_{DAAM}$, the source node A_n forwards packets to its parent node. Otherwise, go to step 3.

Step 3. If $A_n \leq A_{DAAM}$ and $D \geq A_{DAAM}$, it calculates the parent node at depth d of the destination node D is the $f(d)^{th}$ child node of its parent, it is given as:

$$f(d) = \begin{cases} \left\lfloor \dfrac{D - A_{DAAM} - M(d)}{C_{offset}(d)} \right\rfloor, & 1 \leq d \leq L_m - 1 \\ 0, & d = 0 \end{cases} \quad (11)$$

Among, $M(d)$ is given as:

$$M(d) = \begin{cases} \sum_{j=0}^{d-1} f(j) \times C_{offset}(j) + d \times \beta, & 1 \leq d \leq L_m - 1 \\ 0, & d = 0 \end{cases} \quad (12)$$

The formula (13) is given as:

$$D - A_{DAAM} - M(d) \leq \beta \quad (13)$$

If the formula (13) is true, then the destination node D is the child node of the parent node at depth d and calculating the address of its parent node using formula (3) and set its parent node is the next hop forwarding node.

4 BAADM ALGORITHM THEORY ANALYSIS

Theorem 1. BAADM and DAAM algorithm has the same complexity in terms of time and memory.

Proof. The performance period of BAADM and DAAM algorithm is mainly affected by the maximum neighbor nodes number N_n and the network maximum depth L_m. Because the maximum neighbor nodes number N_n and the network maximum depth L_m of BAADM and DAAM algorithm are the same, So their time complexity is $O(L_m + N_n)$. The main store space of BAADM and DAAM algorithm is occupied by the information about neighbor nodes, so their storage complexity is determined by the maximum neighbor nodes number N_n. Because the maximum neighbor numbers of BAADM and DAAM algorithm is N_n, so their memory complexity is the same, which is $O(N_n)$.

Theorem 2. Tree routing algorithm based on BAADM and DAAM algorithm have the same maximum number of routing hops which is $2L_m$.

Proof. In the tree routing algorithm, $LCA(A_n, D)$ is the minimum depth common parent node of source node A_n and destination node D according to address assignment mechanism. $level(u)$ is used to show level of node u. $HopCount$ shows the tree routing hops from source node to destination node, it is shown in formula (14) [8]:

$$HopCount = level(A_n) + level(D) - 2 \times level(LCA(A_n, D)) \quad (14)$$

If the network depth of node A_n and node D in different branches are L_m, then their common ancestors node with the minimum depth is coordinator. The maximum network depth of the tree routing algorithm based on BAADM and DAAM is L_m, so tree routing algorithm based on BAADM and DAAM algorithm have the same maximum number of routing hops which is $2L_m$.

5 SIMULATION AND ANALYSIS

In order to evaluate the performance of BAADM algorithm, using NS2 simulation platform. We simulate BAADM, DAAM, SLAR, SOSAA and DAAM-THN algorithm in the different node number respectively. Coordinator is located in the center of the network, other nodes random distribution outside of the coordinator and configures $C_m = 5$, $R_m = 3$, $L_m = 8$. The experiment data is the average of running the algorithm 50 times independently, the parameters are shown in table 1.

Table 1. The simulation parameter settings

Parameter	Values
Network range /m $\times$ m	100×100
Number of nodes	100，200, …, 500
Communication radius /m	35
The proportion of end nodes	40%
The proportion of routing nodes	60%

5.1 *The success ratio of address assignment*

The success ratio of address assignment is given as:

$$S = k_s / k \qquad (15)$$

Where k = the total number of nodes; and k_s = the number of nodes that have obtained addresses.

Fig. 1 shows BAADM algorithm in different node number scenarios, the values of S were greater than DAMM, SLAR, DAAM-THN and SOSAA algorithm. The reason for this is that BAADM algorithm under the premise of not affect DAAM algorithm, makes more nodes obtain the network address.
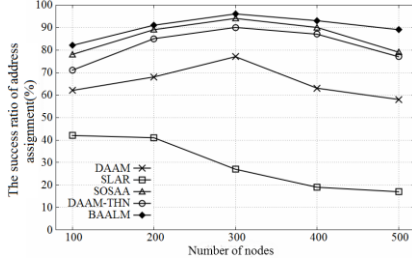

Figure 1. Compare of the success ratio of address assignment

5.2 *The average time of address assignment*

The average time of address assignment is given as:

$$t = t_s / k_s \qquad (16)$$

Where t_s = the total time of assigning addresses. Fig. 2 shows that the value of t for BAADM algorithm is less than DAAM and SLAR algorithm on the whole. In the smaller node numbers, the value of t is basically the same. But along with the increase of node numbers, the value of t will be reduced. The reconfiguration process of SLAR algorithm and the two hop communication of DAMM-THN algorithm have made the value of t increased significantly.
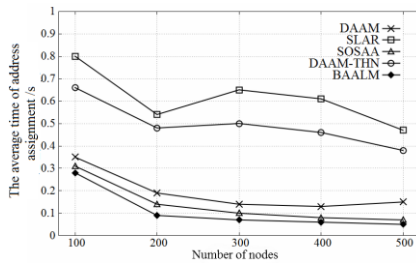

Figure 2. Compare of the average time of address assignment

5.3 *The maximum depth of the network is influence to the success ratio of address assignment*

The scene choose 500 nodes, C_m=5，R_m=3，L_m= {2，3，4，5，6，7，8，9}. Fig. 3 shows that BAADM algorithm has a better success ratio of address assignment. Because of DAAM algorithm defined address space enlarge, so the extensible address space of SOSAA algorithm will reduce, and the success ratio of address assignment will drop. In the case of the same computational complexity, BAADM algorithm is less affected by the maximum depth L_m of the network.
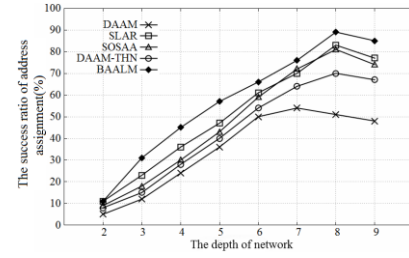

Figure 3. The maximum depth of the network is influence to the success ratio of address assignment

6 CONCLUSIONS

In this paper, we have been proposed the BAADM algorithm to improve the existing borrowed address assignment algorithms. On the basis of SOSAA algorithm, we fully consider that the network address waste is influence to address allocation performance. Theoretical analysis and simulation results show that, in the condition of the same computational complexity and the same maximum network depth, the BAADM algorithm outperforms DAAM, SLAR, DAAM-THN and SOSSA in terms of the success rate of address assignment, and the average time of assigning address etc.

REFERENCES

[1] ZigBee Alliance. ZigBee specification[S].2008.

[2] Unoma Ndili, Okorafor. & Kundur, Deepa. 2009. On the Relevance of Node Isolation to the K-Connectivity of Wireless Optical Sensor Networks[J]. IEEE Transactions On Mobile Computing 8(10): 1427-1440.

[3] Debabrato, Giri. & Uttam Kumar, Roy. 2009. Single level address reorganization in wireless personal area network[C]. 2009 -4th International Conference on Computers and Devices for Communication. Kolkata, India, 14-16 December 2009: 1-4.

[4] Zhi, Ren. & Pengxiang, Li. & Yukun, Yao. & Yong, Huang. 2012. Segmentation-based on-demand scalable address assignment algorithm in ZigBee networks[J]. Journal on Communications 33(5): 131-137.

[5] Xujun, Wu. & Yongli, Wang. 2013. Borrowed Address Assignment Algorithm in ZigBee Networks Based on Two-hop Neighborhood[J]. Science Technology and Engineering 13(28): 8333-8338.

[6] Yukun, Yao. & Guan, Wang. & Zhi, Ren. & Pengxiang, Li. & Yongchao, Chen. 2013. Efficient distributed address assignment algorithm based on topology maintenance in ZigBee networks[J]. The Journal of China Universities of Posts and Telecommunications 20(3): 53-59.

[7] Mostafa K.A.Al-Harbawi. & Mohd Fadlee A. Rasid. & Nor Kamariah Noordin. 2012. Utilizing Neighbours-Table to Improve Tree Routing Protocol in ZigBee Network[J]. Wireless Personal Communications 65(2): 469-488.

[8] Taehong Kim. & Seong Hoon Kim. & Jinyoung Yang. & Seong-eun Yoo. & Daeyoung Kim. & Member, IEEE. 2014. Neighbor Table Based Shortcut Tree Routing in ZigBee Wireless Networks[J]. IEEE Transactions on Parallel and Distributed Systems 25(3): 706-716.