Hybrid Distributed Stochastic Addressing Scheme for ZigBee/IEEE 802.15.4 Wireless Sensor Networks

Hyung Seok Kim and Jiwon Yoon

This paper proposes hybrid distributed stochastic addressing (HDSA), which combines the advantages of distributed addressing and stochastic addressing, to solve the problems encountered when constructing a network in a ZigBee-based wireless sensor network. HDSA can assign all the addresses for ZigBee beyond the limit of addresses assigned by the existing distributed address assignment mechanism. Thus, it can make the network scalable and can also utilize the advantages of tree routing. The simulation results reveal that HDSA has better addressing performance than distributed addressing and better routing performance than other on-demand routing methods.

Keywords: Distributed stochastic addressing, IEEE 802.15.4, ZigBee, wireless sensor network, routing.

Manuscript received Aug. 23, 2010; revised Feb. 7, 2011; accepted Feb. 22, 2011.

This research was supported by the Ministry of Knowledge Economy (MKE), Rep. of Korea, under the Convergence Information Technology Research Center (Convergence-ITRC) support program (NIPA-2011-C6150-1101-0003) supervised by the National IT Industry Promotion Agency (NIPA), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the MEST (2010-0025316).

Hyung Seok Kim (+82 70 7137 1237, email: hyungkim@sejong.ac.kr) and Jiwon Yoon (email: vincer@naver.com) are with the Department of Information and Communication Engineering, Sejong University, Seoul, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.11.0110.0501

I. Introduction

As wireless sensor networks (WSNs) are expected to assume a greater role in everyday life, ZigBee, the representative communication network protocol of WSN, has become an active research subject. ZigBee is a low-power wireless protocol based on MAC/PHY of the IEEE 802.15.4 standard [1].

In order to efficiently construct a WSN consisting of many sensor nodes, it is necessary to consider addressing and routing algorithms. The ZigBee standard [2] specifies two addressing methods—the distributed address assignment mechanism (DAAM) of a hierarchical tree structure and the stochastic addressing assignment mechanism (SAAM), which assigns addresses randomly. DAAM performs hierarchical addressing and routing with minimum memory use and routing overhead by appointing the maximum tree depth (Lm), maximum number of children (Cm), and maximum number of routers (Rm) in advance. SAAM allows for network scalability using random addresses. Using a routing table with the assigned addresses, it searches routes in a manner similar to the ad-hoc on-demand distance vector (AODV).

DAAM has characteristics that make it suitable for a WSN; for example, it uses few memory resources in routing and it is simple. However, it may restrict the entry of new nodes before exhausting all available addresses by limiting hop counts and the number of children. When (Cm, Rm, Lm) is (5, 5, 7), theoretically, at most 97,655 (> 65,535) addresses are available. When nodes are placed around the first personal area network (PAN) coordinator in the network, no one knows whether the nodes around the PAN coordinator become child nodes of the PAN coordinator or will form their linear chain. If the latter takes place, the network area can only be narrowed because

Lm is quickly consumed. When a newly entering node requests address assignment from a parent node, the parent node may not assign an address to the node due to the limit of $Lm = 7$ or $Cm = 5$. If nodes enter from unexpected directions, the problems will be more serious. The higher the number of surrounding nodes, the more frequently this problem will occur. When using DAAM, a detour may also occur in the route between nodes.

SAAM can form a scalable network and construct short routing paths. However, because SAAM may duplicate addresses, it must perform duplicate address detection [3] as a preventative countermeasure. In addition, it needs to use relatively greater memory compared to the limited capacity of the sensor nodes as a result of the routing table and algorithm [4].

There have been studies regarding addressing of ad-hoc networks. First, in the centralized addressing method [5], [6], a central node stores, manages, and assigns network addresses for all nodes in the network. However, in this case, if there is an error with the central node, the entire network would not work. Moreover, the central node might get overburdened. Another method is the distributed addressing method [7]-[9], which has difficulty managing addresses and must check for duplicate addresses.

Several studies have been performed on the addressing of ZigBee networks. In [10], graph algorithms were proposed to alleviate the problem of DAAM. However, it does not suggest how to allocate addresses to nodes when they join the network. Its algorithms consider how to reconstruct the tree network to decrease orphan nodes by the iterative processing. Centralized addressing by the ZigBee coordinator and hybrid addressing utilizing both DAAM and centralized addressing were proposed in [11]. However, the centralized addressing may incur too long delay in multihop WSNs and point of failure.

The distributed addressing proposed in [12] is a compound scheme combining DAAM and the addressing method utilizing prime numbers. We focus on the distributed addressing in this paper, thus hybrid address assignment (HAA) [12] is compared to the proposed method. In distributed borrowing addressing (DIBA) [13], when a new node cannot enter the network because of topological parameter limits of DAAM, an address can be assigned to the new node by borrowing one of addresses that the neighbor node does not assign yet to its child nodes. This method still cannot use all unreserved addresses like DAAM.

This study presents the development of hybrid distributed stochastic addressing (HDSA) as an efficient distributed addressing method. It utilizes the advantage of DAAM and SAAM and overcomes the disadvantages. A routing algorithm suitable for HDSA is proposed to save resources in WSNs.

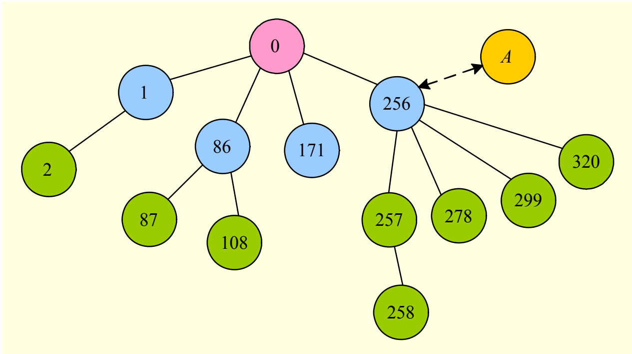

Fig. 1. Entry of new nodes into the (4, 4, 4) network.

Section II describes how to address sensor nodes and route using HDSA, the proposed address assignment method. Section III compares the proposed method with the existing addressing and routing algorithms and evaluates its performance. Section IV ends this paper with conclusions and directions for future work.

II. Hybrid Distributed Stochastic Addressing

1. Address Assignment

Sensor nodes search for parent nodes for association through scanning when new nodes attempt entry in the ZigBee network. In DAAM, addresses are assigned from the parent nodes through an addressing method called *Cskip* when *Cm*, *Rm*, and *Lm* have not reached their limits. However, when *Cm*, *Rm*, or *Lm* is over the limit, a new node fails to join the network.

According to the addressing method of ZigBee specification [2], the maximum address number A_{max} in the network is

$$A_{max} = Cskip(0) \times Rm + Cm - Rm, \tag{1}$$

where

$$Cskip(0) = \begin{cases} 1 + Cm \cdot (Lm - 1), & \text{if } Rm = 1, \\ \dfrac{1 + Cm - Rm - Cm \cdot Rm^{Lm-1}}{1 - Rm}, & \text{otherwise.} \end{cases} \tag{2}$$

For example, when (*Cm*, *Rm*, *Lm*) is (4, 4, 4), the maximum address number A_{max} is 340 according to the address calculation equations $Cskip(0) \times Rm + Cm - Rm$ and $Cskip(0) = 85$ [8]. Thus, one cannot use most addresses among 65,535, the maximum available number of addresses of 16 bits. Addresses that are not used from 0 to 340 are called unused addresses, and ones not reserved from 341 to 65,535 are called unreserved addresses.

In the example of the (*Cm*, *Rm*, *Lm*) = (4, 4, 4) network in Fig. 1, when a new node *A* cannot enter the network because of *Cm* limits in DAAM, addresses can be assigned to new nodes

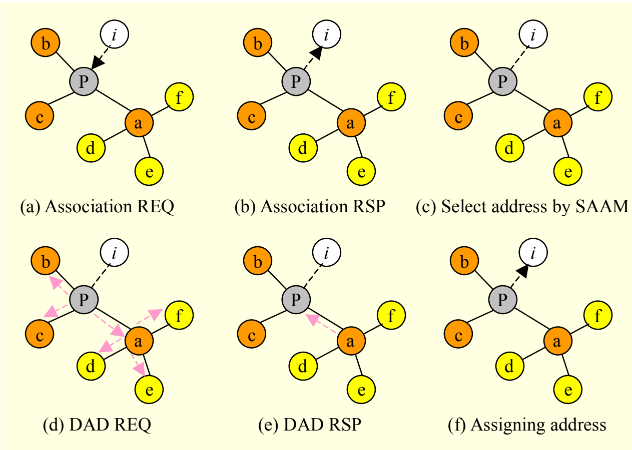

Fig. 2. Process of assigning addresses by HDSA.

by borrowing unused addresses using a method described in [13]; however, it still cannot use all unreserved addresses.

Using the proposed HDSA in Fig. 1, a new node *A* finds node 256 through scanning and attempts to enter the network. Node 256 already has four child nodes and cannot receive further child nodes because of the *Cm* limit. Thus, the HDSA randomly selects one of the unreserved addresses from 341 to 65,535, assigns it to the new node, and has it associated as a child node. HDSA allows a network to continue expanding by assigning addresses randomly with SAAM for the nodes for which address assignment is not possible with DAAM.

When a new node tries to enter the network, it will receive a random address if the parent node has one. That is, if the candidate parent node has an address assigned through SAAM, its child nodes will have addresses assigned through SAAM. In such a case, the parent node, which received addresses in DAAM, will serve as the cluster head and will form a cluster with the nodes whose addresses are assigned through SAAM. The nodes added through SAAM are to be assigned in the locations that exceed the *Cm*, *Rm*, and *Lm* limits and would therefore be few in number. The nodes using SAAM are managed by the nodes whose addresses are assigned through DAAM, which implies that they can reduce the amount of routing overhead compared to entirely SAAM-based networks.

Performing duplicate address detection (DAD) dealing with all nodes may create considerable overhead. HDSA allows the cluster heads that are the DAAM nodes to perform DAD for the subordinate SAAM nodes in the tree, thereby reducing DAD overhead considerably.

Figure 2 illustrates the process of assigning addresses by HDSA. When a new node tries to enter the network, it looks for nodes to which it can request a connection through scanning. In Fig. 2, a new node *i* requests node *P* for a connection. Upon receiving the request for address assignment, node *P*, a parent node, assigns a random address in order to solve the limited number of children nodes according to the number of *Cm*. However, assigning random address may cause the problem of duplicate addresses because addresses are not stored in the central node, that is, the PAN coordinator. In order to solve this problem, HDSA performs DAD before assigning the address. Before assigning the address, the parent node forwards a DAD REQ message within the tree structure. Upon receiving the message, the cluster head nodes on the tree search the tables for node management within their cluster and reply if they have the same addresses. When detecting duplicate addresses, node *P* once again selects a random address and goes through DAD. When there is no duplicate address, it will assign the selected address to node *i*, which is a new entering node. In this manner, the node will finally enter the network.

2. Routing Algorithm

The basic structure of a network continues to have a tree topology when the network is constructed with HDSA. As HDSA is the combination of DAAM and SAAM, a new routing protocol for HDSA is proposed by combining the routing protocols of the two methods.

A network example using HDSA is presented in Fig. 3 to explain routing protocols. In the network (*Cm*, *Rm*, *Lm*) = (4, 4, 4), nodes *p* and *r* receive random addresses 25,000 and 54,040, respectively. A node *q* that is the child of node 25,000 is also assigned a random address 35,400. Because the tree structure can be used as a kind of backbone network, a simple tree routing algorithm with the cluster heads is proposed for HDSA. The basic principle of HDSA is that tree routing is used for nodes within the tree structure and AODV routing is performed for randomly assigned addresses. There are four types of routes according to the addressing methods of source and destination nodes. Addresses from 0 to A_{max} are ones for DAAM and addresses from A_{max} + 1 to 65,535 are for SAAM. In this manner, each node can easily recognize the type of the address of destination node by the address.

Case 1. DAAM to DAAM (for example, node 2 to node 320 in Fig. 3)

In this case, data packets are delivered between the nodes in the tree structure based on DAAM of ZigBee specification [2]. The hierarchical tree routing method based on *Cskip* judges whether or not the destination is a descendant and forwards packets according to

$$Cskip(d) = \begin{cases} 1 + Cm \cdot (Lm - d - 1), & \text{if } Rm = 1, \\ \dfrac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm}, & \text{otherwise}, \end{cases} \quad (1)$$

d: depth of network.

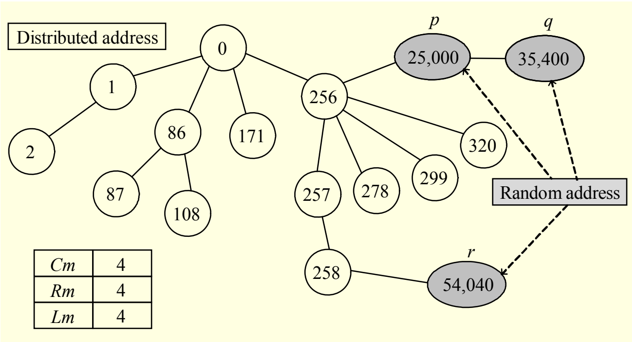

Fig. 3. Example of address assignment using HDSA.

Case 2. SAAM to DAAM (for example, node 35,400 to node 2 in Fig. 3)

This case is applicable when the node that receives an address through SAAM is the source node and the one within the tree is the destination node. Upon checking that the destination address of the packet is the one assigned through DAAM, the source keeps forwarding packets to the parent node until it reaches the node of DAAM address, the backbone node. When the packet sent by node 25,000 arrives at node 256, node 256 delivers the packet to the destination node like the DAAM to DAAM method explained in case 1.

Case 3. DAAM to SAAM (for example, node 1 to node 25,000 in Fig. 3)

This is the case in which the source node is assigned through DAAM and the destination address is assigned through SAAM. When assigning an address through SAAM, one cannot use the tree routing according to *Cskip* calculations. Thus, the source node floods the ROUTE_REQ message to all nodes of the tree. The nodes that keep the address of the destination node in the table among the DAAM-based nodes on the tree will respond through the ROUTE_RSP message. The node that keeps the destination address in the table and is located on the tree becomes the intermediate destination node, and the delivery is made to the final destination address included within the packet. Packets are sent to the DAAM node as the intermediate destination through tree routing. Subsequently, the node and cluster members will continue to forward the packets to the nodes connected to them until the packets reach the final destination.

There are two ways to deliver data in clusters with SAAM nodes. One way is to use a routing table, and the other is to use the broadcasting method. Using a routing table, one applies AODV between the SAAM nodes under the DAAM nodes. The broadcasting method needs no additional routing algorithm. One of these aforementioned ways is selected, considering the number of SAAM nodes that the cluster head manages or the memory capacity of the node. The routing table of DAAM node has information on HDSA nodes which are descendants of DAAM node. The routing table has 16-bit addresses of HDSA nodes as destination nodes and 16-bit address of the next-hop node to which the packet should be forwarded for each destination, as listed in Table 1. This means that the storage of DAAM node for the routing table costs 32 bits per one HDSA node. Average storage cost per node for the routing table can be obtained. It is different with each setting of (*Cm*, *Rm*, *Lm*) and is listed in Table 2.

Table 1. Node 256's routing tables of nodes in Fig. 3.

	Address	Next hop addr.
Dest1	25,000	25,000
Dest2	35,400	25,000

Table 2. Average storage cost per node for routing table.

(*Cm*, *Rm*, *Lm*)	Average storage for routing table	(*Cm*, *Rm*, *Lm*)	Average storage for routing table
(3, 3, 9)	39 bits	(6, 6, 6)	5 bits
(4, 4, 7)	64 bits	(7, 7, 5)	75 bits
(5, 5, 6)	75 bits	(8, 8, 5)	24 bits

```
If(dest address is my address) then
    Get the packet
Else If(dest address belongs to child's block) then
    If(my address is assigned by SAAM or my address is
    cluster head) then
        Forward the packet using AODV within the cluster
    Else
        Forward the packet to the child
    End if
Else
    Forward the packet to the parent
End if
```

Fig. 4. Routing algorithm for HDSA.

Case 4. SAAM to SAAM (for example, node 54,040 to node 35,400 in Fig. 3)

This case is applicable when data are delivered from the nodes containing SAAM addresses to other SAAM address nodes connected to DAAM clusters. It performs routing by combining SAAM to DAAM, as described in case 2, and DAAM to SAAM, as described in case 3. In the example of the figure, node 54,040 sends the ROUTE_REQ message to the DAAM node. Node 320 with a DAAM address floods it within the tree. Node 256 with the concerned address then sends ROUTE_RSP to node 54,040. Upon receiving ROUTE_RSP, node 54,040 sends the packet that indicates that node 256 is the intermediate destination and node 35,400 is the final destination. The packets delivered to an upper node from node 54,040 meet node 320, a DAAM node, and the delivery

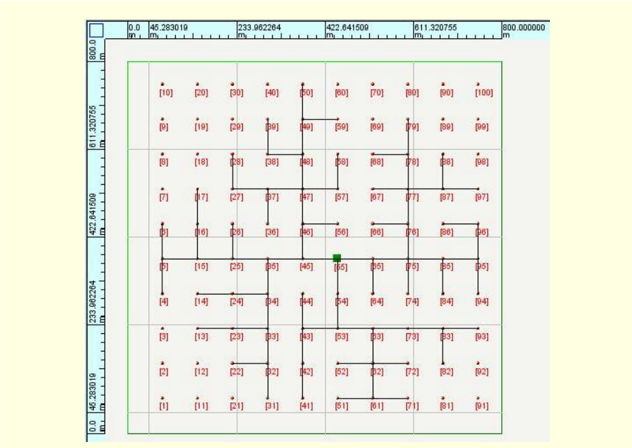

Fig. 5. Example of DAAM addressing (*Cm*, *Rm*, *Lm*) =(3, 3, 6).

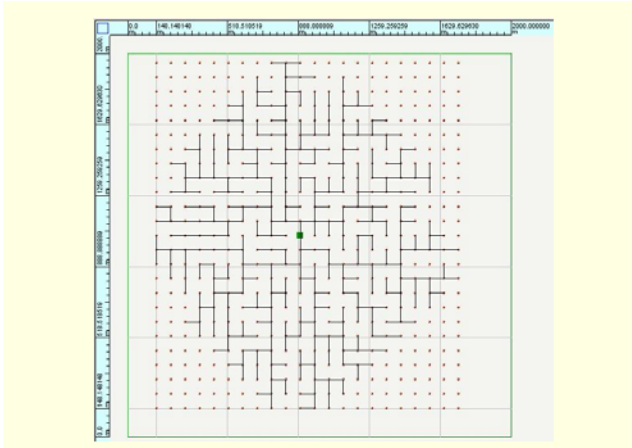

Fig. 6. Example of node arrangement using DAAM (2, 2, 14).

Table 3. Number of maximum addresses that can be used by DAAM.

(*Cm*, *Rm*, *Lm*)	Number of addresses	(*Cm*, *Rm*, *Lm*)	Number of addresses
(3, 3, 9)	29,523	(6, 6, 6)	55,986
(3, 3, 10)	88,572	(6, 6, 7)	335,922
(4, 4, 7)	21,844	(7, 7, 5)	19,607
(4, 4, 8)	87,380	(7, 7, 6)	137,256
(5, 5, 6)	19,530	(8, 8, 5)	37,448
(5, 5, 7)	97,655	(8, 8, 6)	299,592

is made with tree routing to node 256, the intermediate destination. Node 256 delivers the packets to nodes within its cluster, using AODV routing, and they finally arrive at the final destination node 35,400.

III. Performance Analysis

1. Address Assignment Rate

HDSA assigns and manages addresses in the SAAM method when sensor nodes find it difficult to have addresses assigned to them when entering a network in the DAAM-based tree structure. Addresses that can be additionally assigned differ according to the initial setting (*Cm*, *Rm*, *Lm*) because it can assign the remaining addresses after excluding the number of addresses available in the DAAM-based tree structure from 65,535 addresses that sensor nodes can have in a ZigBee network.

For example, let us assume that there is a network (*Cm*, *Rm*, *Lm*) = (3, 3, 7). The number of addresses that new nodes can have is calculated to be 2,047. These 63,488 (= 65,535 − 2,047) nodes can have addresses assigned to them in SAAM.

Figure 5 presents the results of assigning and arranging addresses to 100 sensor nodes through DAAM (*Cm*, *Rm*, *Lm*) = (3, 3, 6) with 75 cm between them within the scope of 800 m ×800 m using a Qualnet simulator. Although addresses can be assigned to a maximum of 2,047 nodes theoretically, 28 of 100 nodes remain unaddressed. When it comes to 16-bit addresses used in the ZigBee network hierarchy, they can have 65,535 addresses. Table 3 presents the maximum number of addresses that can be used in DAAM according to the *Cm*, *Rm*, and *Lm* parameters. As listed in Table 3, the use of limited parameters close to 65,535, which is the number of maximum addresses used by ZigBee, is correct when using DAAM.

In HDSA, one can use the unreserved addresses between the number of addresses in Table 3 and the maximum 65,535 and assign addresses to the nodes that cannot have addresses assigned to them through DAAM. HDSA can also assign addresses to the nodes entering the network if the (*Cm*, *Rm*, *Lm*) parameters are set to allow for enough unreserved addresses.

Large (*Cm*, *Rm*, *Lm*) parameters decrease the number of addresses for SAAM. Figure 6 presents an example of a sensor network using (*Cm*, *Rm*, *Lm*) = (2, 2, 14). A total of 625 nodes are arranged with 75 m distance between each other around the PAN coordinator. Because nodes are addressed in a distributed manner, a node cannot determine which addresses other nodes are assigned. Therefore, although a maximum of 32,766 addresses can be assigned using parameters (2, 2, 14), all 625 nodes arranged in the figure may not have the addresses assigned to them. As shown in Fig. 6, the sensor network that intends to arrange 625 nodes can fail to use 32,766 addresses in the tree structure and assign addresses to only 355 nodes. When the nodes without addresses assigned to them receive addresses through SAAM, all 625 nodes will have addresses assigned to them.

In DAAM, nodes may belong to the children of the next-

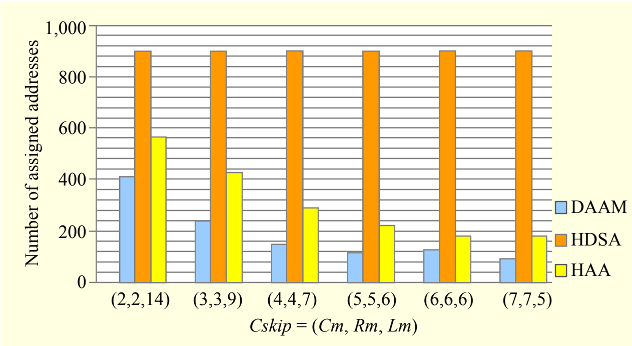

Fig. 7. Comparison of address assignment according to (*Cm*, *Rm*, *Lm*) changes.

Table 4. Comparison of address assignment rate for 65,535 nodes.

(*Cm*, *Rm*, *Lm*)	Type	Address assignment rate (%)
(2, 2, 14)	DAAM	0.6
	HDSA	50
(3, 3, 9)	DAAM	0.3
	HDSA	40.05
(4, 4, 7)	DAAM	0.227
	HDSA	66.89
(5, 5, 6)	DAAM	0.175
	HDSA	70.37
(6, 6, 6)	DAAM	0.1922
	HDSA	14.76
(7, 7, 5)	DAAM	0.137
	HDSA	70.22

depth node before having as many children nodes as *Cm*. Thus, many nodes are left unaddressed even though the final child node of the tree has the same depth as *Lm*. When using DAAM, nodes appear that fail to receive addresses in this way. Because HDSA assigns addresses with unreserved addresses, nodes of the network can have addresses assigned to them as long as unreserved addresses remain. Figure 7 presents the comparison results of HDSA, DAAM, and HAA in terms of the address assignment rate of 900 nodes.

Table 4 lists the address assignment rate according to (*Cm*, *Rm*, *Lm*) changes when the distance between nodes is maintained at 75 m and DAAM and HDSA are applied to 65,535 sensor nodes. When the parameters (*Cm*, *Rm*, *Lm*) are (2, 2, 14), DAAM assigns addresses to 411 nodes and HDSA to 33,280 nodes; when they are (3, 3, 9), DAAM assigns addresses to 237 nodes and HDSA to 26,249 nodes; at (4, 4, 7), DAAM to 149 nodes and HDSA to 43,840 node; at (5, 5, 6), DAAM to 115 nodes and HDSA to 46,120 nodes; at (6, 6, 6), DAAM to 126 nodes and HDSA to 9,549 nodes; and finally, at (7, 7, 5), DAAM assigns addresses to 90 nodes and HDSA to 46,018 nodes.

Table 4 also lists the dropping address assignment rate to nodes according to the dropping *Lm* in DAAM. Meanwhile, HDSA records the highest 70.37% at (5, 5, 6) and the lowest 14.76% at (6, 6, 6). At (5, 5, 6), the number of usable addresses is the lowest 19,530, and thus, the number of addresses that can be assigned through SAAM is the highest. At (6, 6, 6), the number of usable addresses is 55,986, and the number of addresses that can be assigned through SAAM is only 9,549. HDSA has different numbers of usable addresses according to the (*Cm*, *Rm*, *Lm*) parameters.

2. Routing

Subsection III.1 discussed how HDSA overcame the disadvantages of DAAM in terms of address assignment rate. This subsection demonstrates that HDSA reduces the overhead of the SAAM-based routing compared to the AODV used in SAAM. For comparison, simulations were carried out to assess the performance of routing from DAAM nodes to SAAM nodes and routing from SAAM nodes to other SAAM nodes.

A ZigBee-based sensor network was constructed for evaluation, using the Qualnet simulator. The routing method when assigning addresses through HDSA in the network was compared with the routing method AODV used in SAAM. Simulations were carried out with 100 nodes in the 800 m × 800 m range, 255 nodes in the 1,200 m × 1,200 m range, and 400 nodes in the 1,600 m × 1,600 m range. Constant bitrate (CBR) traffic was transmitted in the simulation over 1,000 seconds; 512 bytes were transmitted every 20 seconds. The distance between all nodes was 75 m. The MICAz energy model [14] was used to measure the energy consumption of the sensor nodes in WSN.

A. DAAM to SAAM

Figure 8(a) presents comparison results of energy consumption when there is traffic in the network. When the number of nodes is 100, the HDSA routing method has 0.7162 mJ energy consumption and AODV 2.397 mJ energy consumption. Because HDSA uses the tree routing within the tree, its routing overhead is smaller than AODV. In fact, energy consumption by HDSA routing accounts for only 30% of that of SAAM routing. At 255 nodes (Fig. 8(b)), HDSA's energy consumption is 1.373 mJ and SAAM's energy consumption is 4.965 mJ, which implies that HDSA routing only uses 28% of the energy consumption by SAAM routing. At 400 nodes (Fig. 8(c)), HDSA uses 2.2307 mJ and SAAM uses 6.96 mJ, which implies that HDSA uses only 32% of energy consumption by SAAM routing. In all three cases of network scale, HDSA

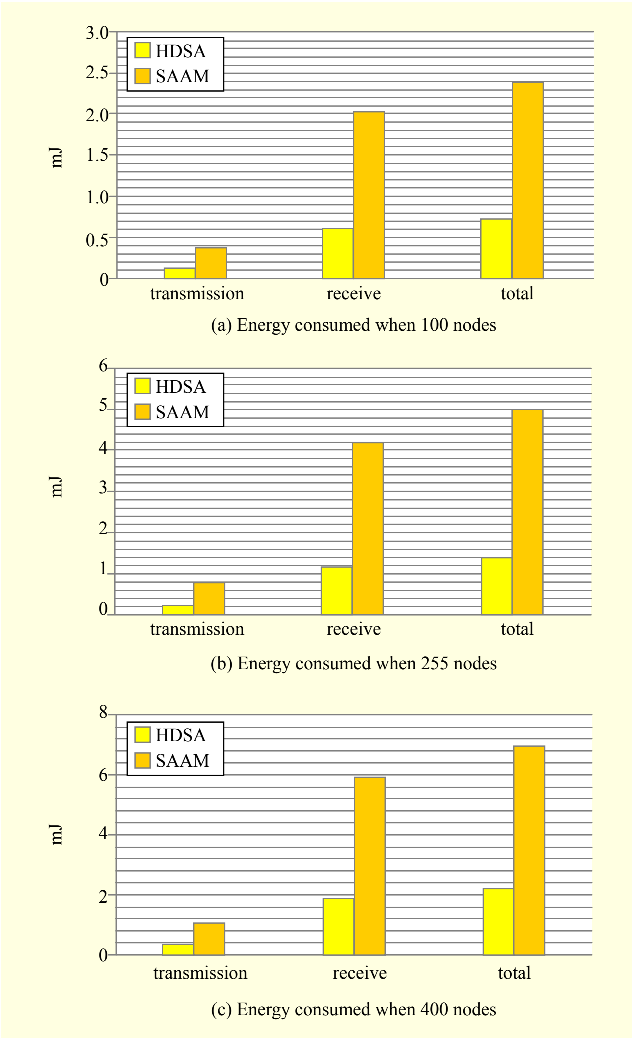

Fig. 8. Energy consumption due to routing when network changes.

consumed less energy than SAAM.

B. SAAM to SAAM

A total of 550 nodes were arranged with 75 m between each. The (*Cm*, *Rm*, *Lm*) parameters changed to (2, 2, 14), (3, 3, 9), (4, 4, 7), and (5, 5, 6). The message overhead when sending messages from an SAAM node to the other SAAM node was compared with AODV. As discussed above, the existing tree routing method delivers packets without separate routing messages. However, separate routing messages are needed when sending to an SAAM-type destination address in HDSA because calculating paths with only random addresses is impossible. Thus, routing request messages should be delivered, and routing responses should be received from, the nodes of SAAM addresses like AODV. Unlike AODV, however, routing request messages within the tree are delivered along the tree. As a result, message overhead decreases when compared to AODV based on the message broadcasting. To evaluate the decrease in overhead, the energy consumption of routing request messages was compared with AODV in each scenario.

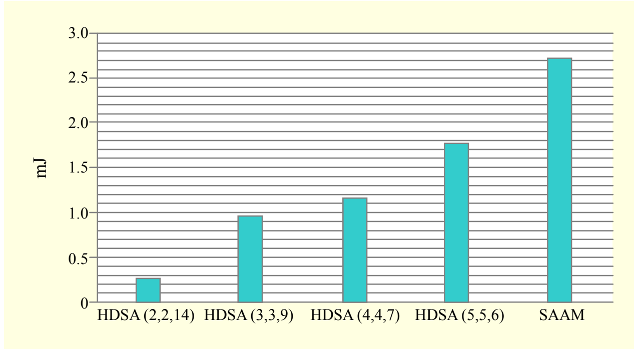

Fig. 9. Energy consumption of HDSA and SAAM during SAAM to SAAM routing.

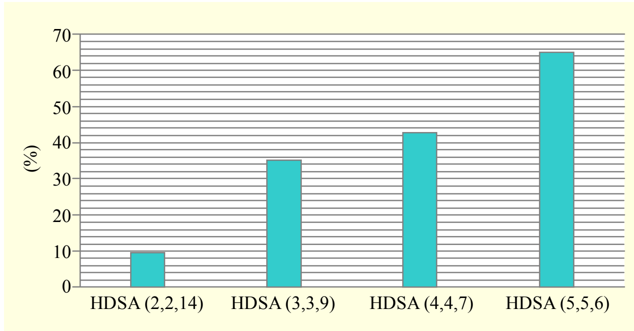

Fig. 10. Routing message overhead ratio of HDSA to SAAM.

Figure 9 illustrates the energy consumption of HDSA and SAAM during routing. The energy consumption, including routing request messages needed for routing, was examined. The CBR traffic, which transmits 50 bytes of data packets every 20 s for 1,000 s, was used. One data delivery is followed by one routing request message delivery. As shown in the Fig. 9, energy consumption when using HDSA is lower than SAAM-based routing. Also, (2, 2, 14) records the lowest energy consumption, which is due to the considerable reduction in energy consumption caused by routing request message delivery within the tree rather than outside the tree. The energy consumption of HDSA was the highest at (5, 5, 6).

Figure 10 shows the energy consumption ratio of HDSA to SAAM. At (2, 2, 14), HDSA consumes 9.9% energy of SAAM; at (3, 3, 9), HDSA consumes 35.06% energy of SSAM; at (4, 4, 7), HDSA consumes 42.68% energy of SAAM; and finally, at (5, 5, 6), HDSA consumes 65.08% energy of SAAM. The network using HDSA considerably reduced the routing message overhead as the size of the tree consisting of DAAM grew. Thus, one can still highlight the advantages of tree routing even when adding extra nodes and expanding the network with HDSA.

IV. Conclusion

This paper proposed HDSA for assigning addresses randomly with unreserved addresses, except those reserved by DAAM, to tackle the problem of being unable to assign addresses to nodes because of the limitation of DAAM, an addressing method of ZigBee. This paper also proposed a routing method suitable for HDSA. Using simulations, HDSA was compared with DAAM in terms of address assignment rate and with SAAM in terms of routing performance. Simulation results revealed that HDSA had a higher address assignment rate than DAAM and lower energy consumption and routing message overhead than SAAM. These results demonstrate that HDSA can provide scalability with relatively low overhead in ZigBee-based WSNs.

References

[1] IEEE TG 15.4b, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE Standard for Information Technology, 2006.

[2] ZigBee Alliance, "ZigBee Specification," Document 05347r17, Jan. 2008.

[3] Z. Yao and F. Dressler, "Dynamic Address Allocations for Management and Control in Wireless Sensor Networks," *Proc. HICSS*, 2007, pp. 292b.

[4] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequence-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM*, 1994, pp. 234-244.

[5] Mesut Günes and J. Reibel, "An IP Address Configuration Algorithm for Zeroconf. Mobile Multi-hop Ad-hoc Networks," *Proc. Int. Workshop Broadband Wireless Ad-Hoc Netw. Services*, Sophia Antipolis, France, Sept. 2002.

[6] S. Toner and D. O'Mahony, "Self-Organizing Node Address Management in Ad-hoc Networks," *LNCS*, vol. 2775, Berlin: Springer Verlag, 2003, pp. 476-483.

[7] K. Weniger and M. Zitterbart, "IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks," *Proc. European Wireless*, Florence, Italy, Feb. 2002.

[8] K. Manousakis et al., "Routing Domain Configuration for More Efficient and Rapidly Deployable Mobile Networks," *Army Sci. Conf.*, Dec. 2002.

[9] P. Patchipulusu, "Dynamic Address Allocation Protocols for Mobile Ad Hoc Networks," Master's Thesis, Texas A&M University, Aug. 2001.

[10] M.-S. Pan, C.-H. Tsai, and Y.-C. Tseng, "The Orphan Problem in ZigBee Wireless Networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 11, Nov. 2009, pp. 1573-1584.

[11] L.-H. Yen and W.-T. Tsai, "The Room Shortage Problem of Tree-Based ZigBee/IEEE 802.15.4 Wireless Networks," *Comput. Commun.*, vol. 33, no. 4, Mar. 2010, pp. 454-462.

[12] Y.-C. Wong et al., "Hybrid Address Configuration for Tree-based Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 12, no. 6, June 2008, pp. 414-416.

[13] S. Park et al., "Distributed Borrowing Addressing Scheme for ZigBee/IEEE 802.15.4 Wireless Sensor Networks," *ETRI J.*, vol. 31, no. 5, Oct. 2009, pp. 525-533.

[14] Shu Chen, Yan Huang, and Chengyang Zhang, "Toward a Real and Remote Wireless Sensor Network Testbed," *Wireless Algorithms, Syst. Appl.: LNCS*, vol. 5258, 2008, pp. 385-396.

Hyung Seok Kim received the BS, MS, and PhD in electrical engineering and computer science from Seoul National University in 1996, 1998, and 2004, respectively. In 2003 and 2004, he held visiting researcher positions in the University of Virginia, USA. From 2004 to 2006, he worked for the Telecommunication R&D center at Samsung Electronics. He is currently a faculty member in the Department of Information and Communication Engineering of Sejong University. His current research interests include cyber-physical systems, wireless sensor networks, and embedded systems.

Jiwon Yoon received the BS and MS in information and communication engineering from Sejong University, Rep. of Korea in 2008 and 2010, respectively. He was a post-graduate student in Department of Information and Communication Engineering of Sejong University when he worked on this paper. His current research interests include embedded systems and network systems.