

# Distributed Borrowing Addressing Scheme for ZigBee/IEEE 802.15.4 Wireless Sensor Networks

Sungjin Park, Eun Ju Lee, Jae Hong Ryu, Seong-Soon Joo, and Hyung Seok Kim

**This paper proposes a distributed borrowing addressing (DIBA) scheme to solve problems of failure in address assignments resulting from limited tree depth and width when the distributed address assignment mechanism is used in a ZigBee/IEEE 802.15.4 wireless sensor network. DIBA is a method of borrowing addresses from neighbor nodes for newly entering nodes and assigning the borrowed addresses. Its network or sensing coverage can increase with almost the same overhead as the existing method. DIBA is a simple and lightweight means of addressing and routing, making it suitable for wireless sensor networks. Simulations showed that DIBA is a distributed addressing scheme with consistently excellent performance.**

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

## I. Introduction

ZigBee is a wireless communication standard for low cost and low power consumption based on IEEE 802.15.4 MAC/PHY [1]. It has been used as a representative protocol for wireless sensor networks (WSNs). The ZigBee standard [2] presents two network address assignment mechanisms: the stochastic address assignment mechanism (SAAM) and the distributed address assignment mechanism (DAAM). In both mechanisms, parent nodes themselves choose addresses and assign them to their children without the help of specific central nodes. This is also characteristic of previously studied mobile ad hoc networks [3]-[5].

SAAM needs to check whether or not addresses are duplicated through duplicated address detection [6] after assigning random addresses. In this case, an on-demand (reactive) [7]-[9] or table-driven (proactive) [10] routing protocol is required when packets are transmitted because addresses are assigned at random. Consequently, these routing protocols are inappropriate for low power/low capacity WSNs because they require frequent broadcasts, large packet header sizes, high memory cost, and so on.

DAAM organizes tree networks and routes packets using only address information, without requests for extra routing tables or route retrieval processes. It guarantees the uniqueness of addresses with a regular address assignment method using a function called Cskip. DAAM begins with previously designated values for the maximum tree depth ( $L_m$ ), the maximum number of children ( $C_m$ ), and the maximum number of children that could be routers ( $R_m$ ). However, when the ultimate aim is a self-organizing WSN [11], [12], it is difficult to expand the network area in a scalable manner using DAAM. For example, when three nodes are placed around the first

---

Manuscript received Feb. 19, 2009; revised June 25, 2009; accepted July 13, 2009.

This work was supported by the IT R&D program of MKE/IITA [2008-F-052, Scalable/Mobile/Reliable Wireless Sensor Network Technology], Rep. of Korea, and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0065891), Rep. of Korea.

Sungjin Park (phone: +82 70 7739 5022, email: parksj@sju.ac.kr) and Hyung Seok Kim (phone: +82 70 7137 1237, email: prof.hskim@gmail.com) are with the Department of Information and Communication Engineering, Sejong University, Seoul, Rep. of Korea.

Eun Ju Lee (email: leeej@etri.re.kr), Jae Hong Ryu (email: jhryu@etri.re.kr), and Seong-Soon Joo (email: ssjoo@etri.re.kr) are with IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

doi:10.4218/etrij.09.0109.0121

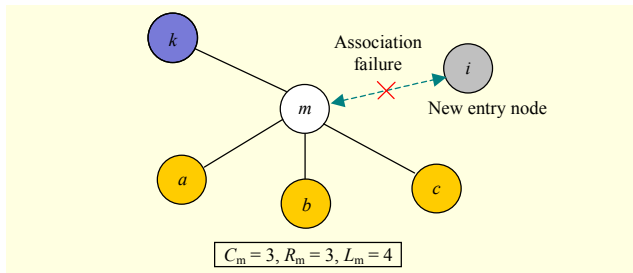


Fig. 1. Address configuration problem of DAAM.

personal area network (PAN) coordinator in the network using  $C_m = 3$ ,  $R_m = 3$ , and  $L_m = 4$ , we cannot know whether the three nodes around the PAN coordinator will all become child nodes of the PAN coordinator or will form a linear chain. If the latter takes place, the network area can only be narrowed because  $L_m$  is quickly consumed. 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 a node enters the network, it is difficult to automatically decide whether it should use  $C_m$  or consume  $L_m$ , given the present condition of address use in its network. Figure 1 shows the problems of DAAM address setup. Newly entering node  $i$  requests address assignment from parent node  $m$ , but node  $m$  cannot assign an address to node  $i$  due to the limit of  $C_m = 3$ .

In [13], there are three solutions for these problems. First, centralized stateful address configuration (CSAC) manages all the addresses in the PAN coordinator. Second, hybrid address configuration assigns addresses using CSAC to the nodes that cannot be assigned addresses based on DAAM. In the third approach, block address configuration, a block of addresses of a certain size is delegated to the router for it to manage. However, they are all centralized methods in which a central node manages addressing. Linear address assignment (LAA) [14] also needs a central node to arrange addressing. LAA requires an additional table for routing, messages for route setup, and an additional header format. It cannot maintain the advantages of DAAM. Centralized addressing may incur a long delay before address assignment because of the need to communicate with the central node, that is, the PAN coordinator. The delay may cause timeout of the association process in the IEEE 802.15.4 standard and cause address assignment to fail.

Consequently, this paper proposes distributed borrowing addressing (DIBA) which maintains the advantages of DAAM and builds a wider network area than DAAM. By borrowing addresses from neighbors, DIBA achieves a higher address assignment success rate and guarantees the uniqueness of addresses. We also propose a routing method for DIBA with a very simple routing table.

The remainder of this paper is organized as follows.

Section II describes the basic operation and routing method of DIBA. Section III compares the performance of the proposed method with DAAM through simulation. Finally, section IV is a conclusion summarizing the study.

## II. Distributed Borrowing Addressing Scheme

### 1. Address Configuration

If a new node enters a ZigBee network, it should find candidate nodes for its parent by scanning. If the newly entering node  $i$  is located within the radio coverage of parent nodes  $m$  and  $c$ , as shown in Fig. 2, node  $i$  selects for its parent the candidate node which has more unoccupied child addresses available for assignment. All router nodes broadcast beacon frames including the number of children that they can add, that is, the number of remaining addresses to assign, called the available address count (AAC), and newly entering nodes can obtain the information from their parent candidates.

The parent node which receives the association request message assigns the address if there remains an available address (AA). On the other hand, if the available address count is zero or the tree depth is  $L_m$  or more, it obtains an address through the address-borrowing mechanism of DIBA. Figure 3 illustrates the operation of DIBA. If node  $m$ , which has received an association request from newly entering node  $i$ , has a limit of 3 children ( $C_m = 3$ ), an address for node  $i$  cannot be assigned because node  $m$  already has nodes  $a$ ,  $b$ , and  $c$  as children. To solve this, node  $m$  should broadcast an address borrowing request (AB\_REQ) messages to borrow addresses from its parent node and children, called neighbor nodes. The neighbor nodes  $k$ ,  $a$ ,  $b$ , and  $c$  respond to node  $m$  with address borrowing response (AB\_RSP) messages including AA and AAC. If AAC is zero, no AB\_RSP message is transmitted. Node  $m$  sets a timeout value and checks the contents of the AB\_RSP messages it receives from neighbor nodes during this interval. The node and address to borrow are selected with the following priority. First, an AA of the node with the biggest AAC is selected. If the AACs are identical, an AA of the

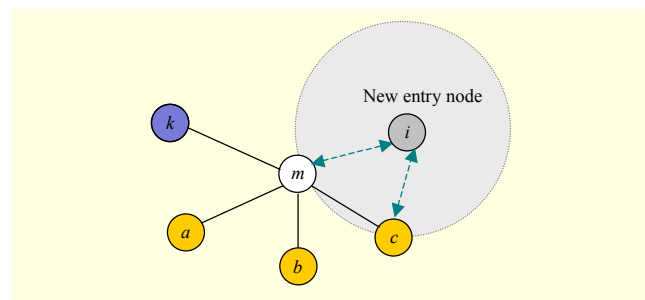


Fig. 2. New node entering a network.

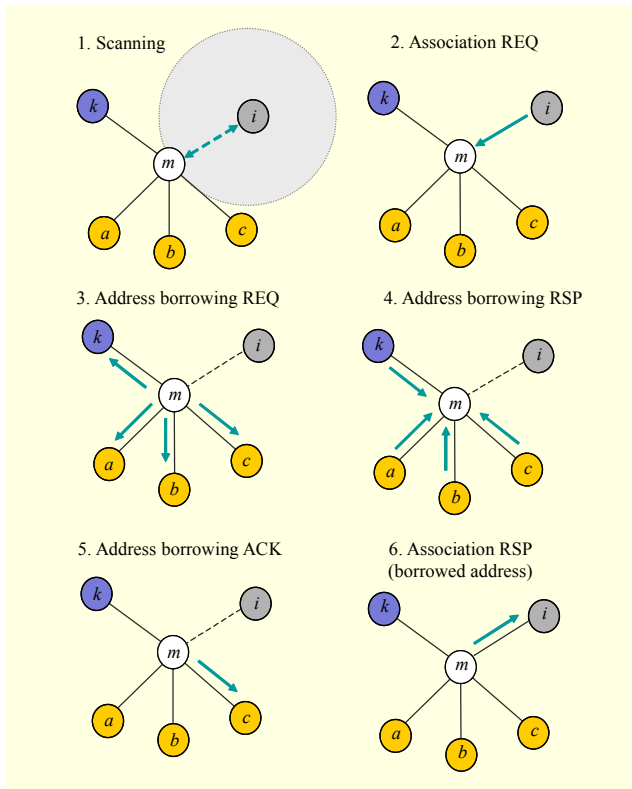


Fig. 3. Address-borrowing mechanism of DIBA.

neighbor node with the highest address should be selected.

If node  $m$  decides to borrow an address from node  $c$  through this process, as shown in Fig. 3, node  $m$  informs node  $c$  through the address borrowing ACK (AB\_ACK) message. On receiving this message, node  $c$  records the address it has lent and updates its routing table. Node  $m$  makes node  $i$  its child by assigning the borrowed address through an association response message. Finally the newly entering node  $i$  joins the network.

According to the IEEE 802.15.4 standard, a device sends the data request command for the association response message to the coordinator  $macResponseWaitTime$  symbols after the acknowledgment of an association request command. The maximum value of  $macResponseWaitTime$  symbols is  $64 \times aBaseSuperframeDuration$ , which is 983 ms at the 2.45 GHz band. In DIBA, after a parent receives the association request, it broadcasts AB\_REQ and waits a certain time for AB\_RSPs. When the timer expires, the parent chooses the AA and then sends an association response command to the new node. Thus, the delay between an association request command and an association response command is the time taken to exchange AB\_REQ and AB\_RSP messages. According to the IEEE 802.15.4 standard, the attribute  $macAckWaitDuration$  is the maximum number of symbols to wait for an acknowledgment frame to a transmitted data frame.

```

If ( $my\_depth < L_m$ ) and ( $num\_child < C_m$ )
  Assign an available address.
  Increase  $num\_child$  by 1.
Else
  Broadcast AB_REQ for the parent and children.
  Set timer (T1).
  While T1 Do
    If (AB_RSPs from all requested nodes have arrived)
      Select the address to borrow using the priority rule.
      Break While.
    End If
  End While
  If (T1 expired)
    Select the address to borrow using the priority rule.
  End If
  Transmit AB_ACK to the lender.
  Assign the AA to the newly entering node.
  Update the routing table.
End If

```

Fig. 4. Algorithm of the parent node.

It is a read-only parameter and is 864  $\mu$ s. Because the data rate of the IEEE 802.15.4 standard is 250 kbps at the 2.45 GHz band, the transmission delay of AB\_REQ and AB\_RSP (less than a size of 40 B) is around 1 ms for each of them. Thus, without loss, it takes less than 3 ms for a node to borrow an address from one of its neighbor nodes. Even in case of packet loss, delay between the association request command and association response command is small enough when compared to the maximum value of  $macResponseWaitTime$ .

In DIBA, a node broadcasts AB\_REQ messages only to its parent node and child nodes within a single hop. If it were to broadcast AB\_REQ to neighbors located over multiple hops, the probability of borrowing an address would increase. However, this might incur a long delay and cause association response timeout. Moreover, if AB\_RSP messages are delivered to many nodes in a dense network, the routing table is complicated and many collisions occur. Thus, because of such overhead, multi-hop borrowing is not appropriate for WSNs. However, if the network area is intended to be more expandable even though overheads are suffered, increasing the hop range for borrowing is possible. Figure 4 is the pseudo-code for the address-borrowing process of DIBA in the parent node.

The nodes that have received AB\_REQ messages seek an appropriate AA to lend using

$$AA = My\_addr + (R_m - num\_BA) \times Cskip(d) + 1, \quad (1)$$

$$num\_BA < R_m$$

where  $Cskip(d)$  is a parameter of DAAM explained in the ZigBee specification,  $d$  is the depth of the current node, and  $num\_BA$  is the number of addresses borrowed. AA is the largest of the addresses which are not already assigned to child nodes or lent to other nodes.

```

If (num_child + num_BA < Cm) and (my_depth < Lm)
  Select AA using (1).
  Transmit AB_RSP including AA.
  Set timer (T2).
  While T2 Do
    If (AB_ACK is received)
      Update the routing table.
      Increase num_BA by 1.
    End If
  End While
End If

```

Fig. 5. Operation of AB\_REQ message-receiving nodes.

If a newly entering node is assigned the available address for lending (AAL) of another node, this means that the node secures as much address space as the address block [AAL, AAL+Cskip( $d$ )-1], where  $d$  is the depth of the node lending the address block. This address block is calculated by the address assignment method of DAAM. Thus, borrowing an address AAL from a node is the same as borrowing as much as the size of the address block [AAL, AAL+Cskip( $d$ )-1]. A part of a borrowed address block cannot be borrowed again. Figure 5 represents the algorithm executed in the nodes receiving the AB\_REQ message as pseudo-code.

If the node which borrows the address leaves the network and the lender is its child, the lender should try to re-associate with another node and be assigned a new address because of the hierarchical tree topology. However, if the lender is the parent of the borrower, the lender is aware that the borrower leaves the network. The lender repossesses the address which was lent to the borrower. If the lender leaves the network and it is the child of the borrower, the borrower does not have to return the borrowed address and keeps the address.

## 2. Routing Algorithm

Figure 6 shows an example ZigBee/IEEE 802.15.4 WSN using DIBA. This network maintains a tree topology as shown in the figure. When a new node tries to associate with node 81, which has three children and cannot have any more children ( $C_m = 3$ ), node 81 borrows the address 104, which is reserved for a child of node 95, from node 95 and assigns the address to node 81's new child. Node 104 in turn accepts nodes 105 and 106 as its children and assigns their addresses.

DIBA uses tree routing based on DAAM and has the advantage of a simple routing algorithm. To show the difference between routings of DIBA and DAAM, let us suppose that node 104 is the destination node, and the source node is located within region A. If the tree routing method of DAAM is used, the data packet is forwarded to node 95 via node 81 because address 104 is originally for a child of node

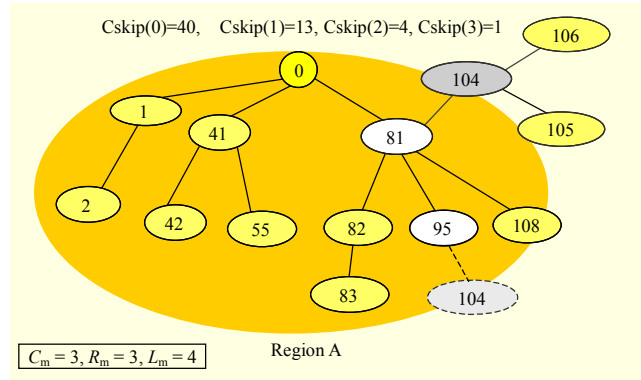


Fig. 6. Example WSN using DIBA.

Table 1. Routing tables of nodes shown in Fig. 6.

(a) Node 81's routing table			
	Address	Lender	Borrower
Parent	0	-	-
Child 1	82	-	-
Child 2	95	-	-
Child 3	108	-	-
Child 4 (borrowed)	104	95	-

(b) Node 95's routing table			
	Address	Lender	Borrower
Parent	81	-	-
Child 1	104	-	81

95. However, the data packet cannot be forwarded to the destination because node 95 lent address 104 to node 81 as shown in Fig. 4. To route packets correctly, in DIBA, each node maintains a routing table which stores the borrowed addresses, lenders, and borrowers, as shown in Table 1. On receiving a data packet, node 81 forwards it to node 104, not node 95, after consulting its routing table.

The routing table maintains the 16-bit short addresses, lender addresses, and borrower addresses as listed in Table 1. A short address is the address of a node and yields the address block owned by the node. The address block includes all the addresses from short address to short address plus Cskip( $d$ )-1 as mentioned earlier. The lender is the node which lends an address, and the borrower is the node which borrows the address. The size of the routing table can be calculated. When a node borrows an address, it adds the borrowed address and the borrower address to its own routing table. Because the size of an address is 2 B, the routing table increases by 4 B whenever it borrows or lends an address. Because an address is borrowed only from a parent or children in DIBA, the additional elements of the routing table are small. Thus, the routing table



```

If (dest address is my address) Then
  Get the packet.
Else If (dest address belongs to a child's block) Then
  If (the block is lent) Then
    Forward the packet to the borrower.
  Else
    Forward the packet to the child.
  End If
Else
  Forward the packet to the parent.
End If

```

Fig. 7. DIBA routing algorithm.

incurs little memory overhead.

Each node performs routing by the algorithm of Fig. 7 based on these three kinds of address information. First, a node that receives a data packet examines whether the destination address is itself or belongs to a child's block. If the destination address belongs to a child's block, the node checks if the child's block has been lent by checking its routing table. If it is lent, the node forwards the packet to the borrower. If it is not lent, the node forwards the packet to the child. If the destination address is not included in the address block of the child nodes, they directly forward the data packets to parent nodes as in the routing method of DAAM. Figure 7 lists the pseudo-code of the routing algorithm.

### III. Simulation

In this section, we evaluate DIBA in ZigBee/IEEE 802.15.4 WSNs using network simulation. The simulation scenarios are divided into two types: uniform and random arrangement of sensor nodes. In the simulations, DIBA was compared with DAAM in terms of the address assignment rate (success rate) and the size of the networking area (sensing area). Simulations were performed using the simulation tool QualNet.

We measured the impact of the tree depth limit ( $L_m$ ) on the address assignment rate and the networking area in a WSN where 900 sensor nodes were uniformly arranged at 100 m spacing in a  $3000\text{ m} \times 3000\text{ m}$  plane. Another simulation was conducted with various  $L_m$  values after 900 sensor nodes were randomly arranged in a  $1500\text{ m} \times 1500\text{ m}$  plane. Second, in order to measure the effect of node density on performance, we increased the total number of nodes by changing the spacing of uniformly arranged sensor nodes from 100 m to 25 m in the regular  $1500\text{ m} \times 1500\text{ m}$  plane. For random arrangement, from 400 to 800 sensor nodes were randomly arranged in the simulation. For various values of  $C_m$ , the performance was evaluated. In all cases, the radio range of the sensor nodes was fixed at 100 m.

The sensor nodes were placed so that less than 70% to 80% of the nodes would have address assignments. The values of

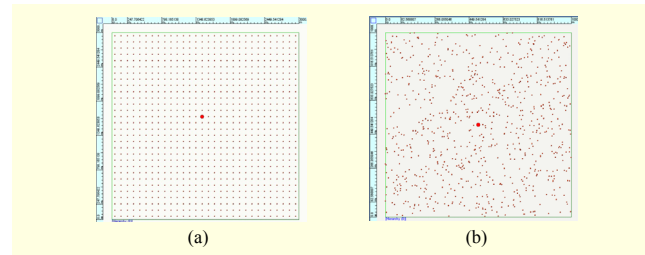


Fig. 8. Node placements: (a) uniform and (b) random.

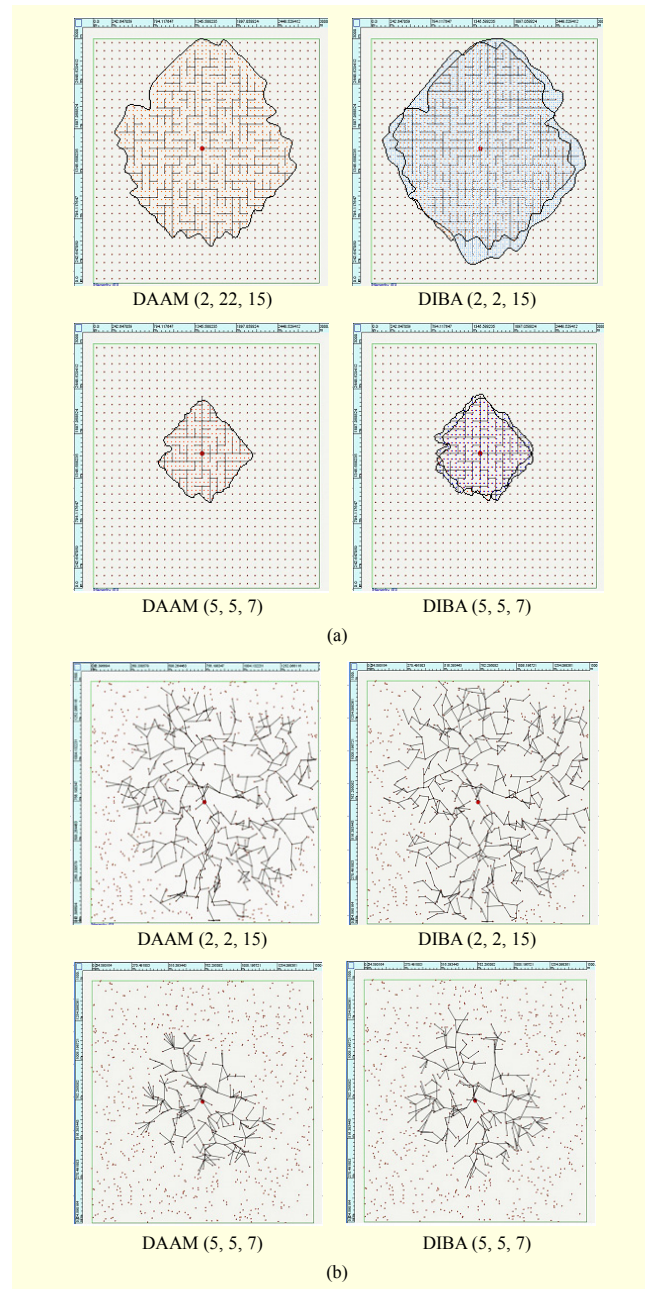


Fig. 9. Tree network links built through simulation of (a)  $3000\text{ m} \times 3000\text{ m}$  area with uniform placement of 900 nodes and (b)  $1500\text{ m} \times 1500\text{ m}$  area with random placement of 900 nodes.

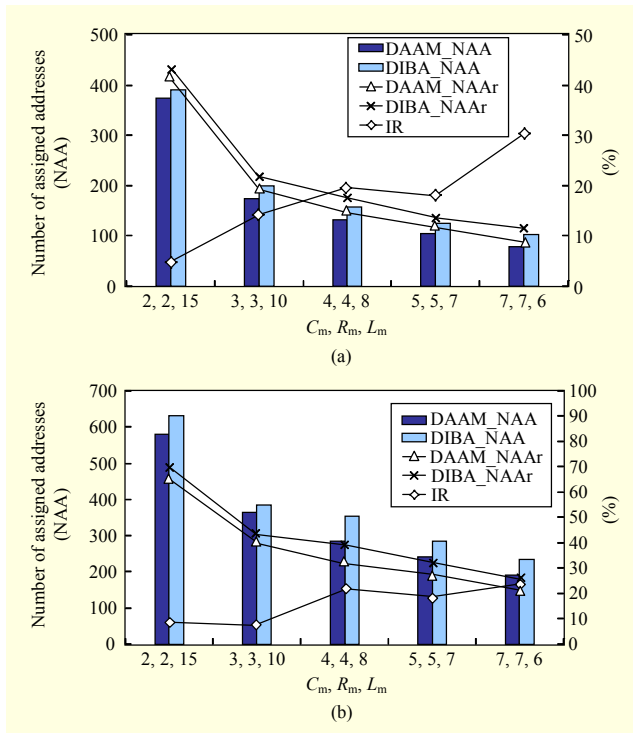


Fig. 10. Impact of  $C_m$ ,  $R_m$ , and  $L_m$  changes on the number of assigned addresses with (a) uniform placement and (b) random placement.

$R_m$ ,  $C_m$ , and  $L_m$  were chosen to ensure that sensors that could not get address assignments always existed. This design was chosen to provide significant changes in assignment rate.

Figure 8 shows uniform and random distributions of nodes for simulation. The big point at the center is a PAN coordinator and the surrounding small points are the coordinators acting as routers. Figure 9(a) shows the network area when using DAAM or DIBA with uniformly arranged sensor nodes, and Fig. 9(b) shows the tree network built through the simulation when sensor nodes were placed randomly.

Figure 10 shows the influence of  $L_m$  on the address assignment rate. The  $x$ -axis shows  $C_m$ ,  $R_m$ , and  $L_m$  in turn. These parameter sets allow assignment of 65,528 addresses. NAA is an abbreviation for *number of assigned addresses* and NAAr is the assignment rate over all 900 nodes. The two bars show the number of addresses assigned by each algorithm, and the two lines represent the assignment rates in percentages. When  $L_m$  decreased from 15 to 7, the graph shows that the number of assigned addresses decreased. DIBA showed a higher success rate than DAAM in address assignment. If the improvement ratio (IR), which is the degree of performance improvement of DIBA over DAAM, is examined, the random node distribution gave the lowest value of 4.8% when  $L_m$  was 10, and gave the highest value of 23.8% when  $L_m$  was 6. The average IR was 16.02%.

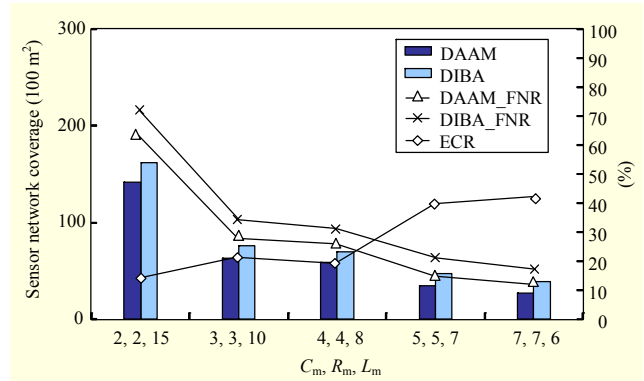


Fig. 11. Impact of  $C_m$ ,  $R_m$ , and  $L_m$  changes on the network coverage.

Figure 11 shows that DIBA widened the sensing coverage (or network coverage) of the sensor network compared with DAAM. The full network coverage ratio (FNR) is the ratio of the area covered by the network to the entire area in the simulation. The two bars show sensing coverage areas in units of 100 m<sup>2</sup> using DAAM or DIBA for various combinations of  $C_m$ ,  $R_m$ , and  $L_m$ . The two lines above the bars show FNR values when DIBA and DAAM were used, respectively. The extended coverage ratio (ECR) is the percentage by which the sensing area was widened by using DIBA instead of DAAM.

As shown in Fig. 11, the higher (lower) the value of  $L_m$  was, the wider (narrower) the network coverage became. DIBA increased FNR values by a maximum of 9% and a minimum of 5% and raised the average by 6% compared with DAAM over all the  $C_m$ ,  $R_m$ , and  $L_m$  combinations. If this is expressed as ECR to show the degree of performance improvement, the values ranged from 14% to 41% with an average of 26%. In the environment where the radio coverage was 100 m and the nodes were uniformly arranged 100 m apart, the sensing area was also widened when the number of nodes involved in the network increased.

Figure 12 shows how node density affected the address assignment rate. In this experiment, NAAr is the ratio of the number of assigned nodes to the maximum number of nodes available for arrangement within the radius of communication when  $L_m$  was 6. In the uniform placement, at most 113, 200, 450, and 1800 nodes could be arranged respectively with node spacings of 100 m, 75 m, 50 m, and 25 m, and these node numbers were used to calculate the percentages. In the experiment with random distribution, the total numbers of nodes shown on the  $x$ -axis of Fig. 12(b) were used for the percentages. DAAM and DIBA results appear in pairs, and the address assignment rate can be seen to increase with larger  $C_m$ .

Because different restrictions ( $C_m$ ,  $R_m$ , and  $L_m$ ) lead to different network topologies, a graph with different restrictions shows irregularities. Note that results of DAAM and DIBA

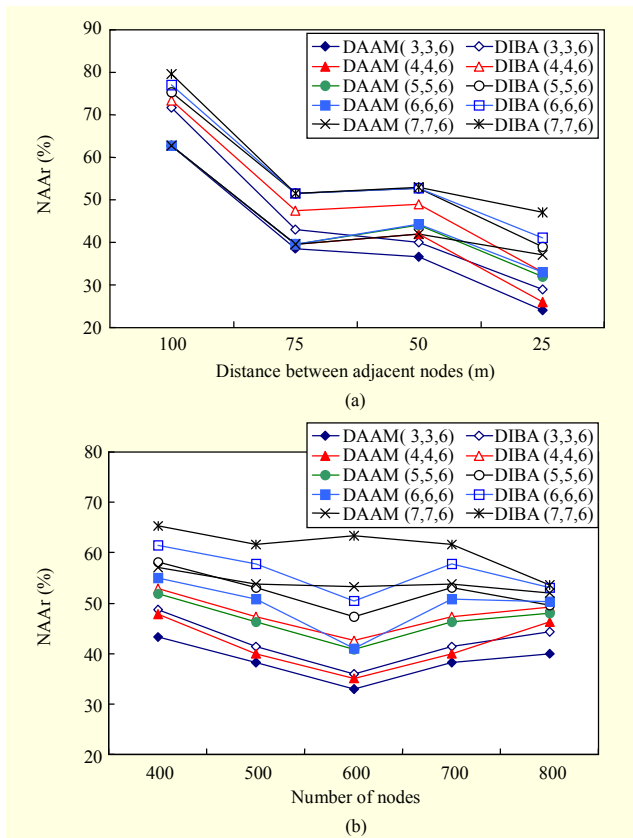


Fig. 12. Address assignment rate according to node density: (a) uniform placement and (b) random placement.

with the same  $C_m$ ,  $R_m$ , and  $L_m$  values should be compared in pairs. DIBA guaranteed higher address assignment rates than DAAM for all  $C_m$  values. The address assignment rates dropped at higher densities in Fig. 12(a) because the denominator of the address assignment rate increased with higher node density while  $L_m$  was fixed. The number of nodes to which addresses were assigned actually increased. DIBA gave higher address assignment rates than DAAM at all values of  $C_m$  and node density as shown in the graphs.

Figure 13 shows how much improvement over DAAM DIBA could give in terms of IR. The performance improvement did not increase regularly with the node spacing as shown in Fig. 13(a). When the node density in the network is normal, nodes which cannot be covered by the maximum length ( $L_m=6$ ) appear and DIBA can improve the address assignment rate over DAAM. In contrast, when the node density in the network is low, many nodes succeed in achieving addresses in DAAM, which leads to a low IR in DIBA. When the node density in the network is high, the denominator of the address assignment rate becomes large, and addresses of many nodes cannot be assigned by DAAM. The numerator of the address assignment rate also becomes larger than that of lower node density. This leads to the irregularity of IR related to the

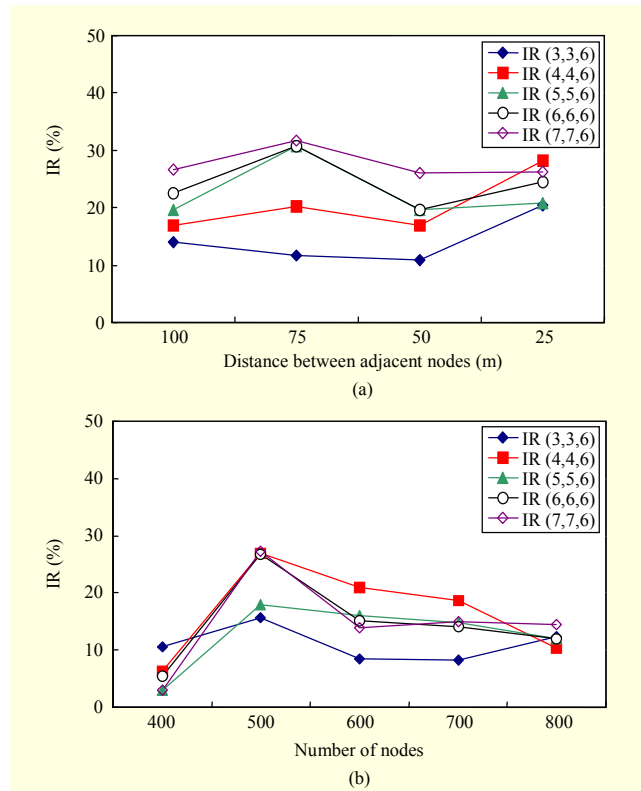


Fig. 13. Improvement rate of DIBA compared with DAAM: (a) uniform placement and (b) random placement.

network topology, which varies according to  $C_m$ ,  $R_m$ , and  $L_m$  as well as the node density. However, almost all the curved lines show average IR values of 15% to 25%. In the random distribution of Fig. 13(b), because the average distances between nodes were largest in the case of 400 nodes, the IR was small. Nonetheless, in the cases of 500 to 800 nodes, the average IR values were from 15% to a little over 20%, except in the case of  $C_m=4$ . DIBA improved the address assignment rate by at least 10% and about 20% on average compared with DAAM.

Figure 14(a) clearly shows the advantages of DIBA and generalizes the previous graphs. The top two lines show the average address assignment rates using DAAM and using DIBA according to the number of nodes in the network. This figure shows the address assignment rates when  $C_m$  was in the range from 3 to 7. DIBA is superior to DAAM in the average address assignment rate. DIBA achieved address assignment rates 7% to 13% higher than those of DAAM. Because DAAM and DIBA guarantee address assignment rates from 30% to 70% in all cases, a 7% to 13% increase in address assignment rates is highly significant. The bottom line shows that IR averaged 21.92%, ranging from 18% to 24%. With random node placement (see Fig. 14(b)), DIBA achieved address assignment rates 7% to 8% higher than those of

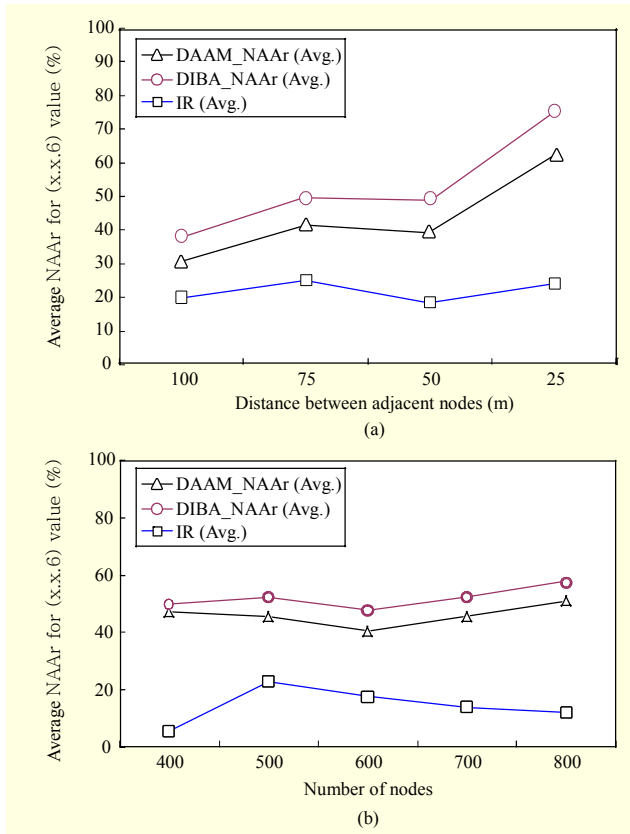


Fig. 14. Average performance comparison: (a) uniform placement and (b) random placement.

DAAM except in the case of 400 nodes. Because the address assignment rates were 40% to 55% in all cases, a 7% to 8% increase is meaningful. The IR averaged 21.92% when the case of 400 nodes was omitted, ranging from 6% to 23%.

#### IV. Conclusion

We proposed DIBA, a method of borrowing unused addresses from neighbor nodes and adopting them. We also suggested a routing algorithm using DIBA. DIBA solves the problem that addresses cannot be assigned to nodes newly entering a network due to limits on the number of children and tree depth when using DAAM, the existing lightweight addressing method of ZigBee. Simulations with both uniform and random node placements were performed using the simulation tool QualNet. We demonstrated the effects of varying the tree depth limit and the node density in the networks. DIBA improved address assignment rates and network coverage compared with DAAM in all the simulated environments. DIBA can expand network coverage and achieve a semi-scalable network using distributed addressing and lightweight routing algorithms suitable for ZigBee/IEEE 802.15.4 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] C.E. Perkins et al., "IP Address Autoconfiguration for Ad Hoc Networks," IETF Internet Draft, draft-ietf-manet-autoconf-01.txt, Nov. 2001.
- [4] S. Nesargi and R. Prakash, "MANETconf Configuration of Hosts in a Mobile Ad Hoc Network," *Proc. IEEE INFOCOM*, 2002, pp. 1059-1068.
- [5] H. Zhou et al., "Prophet Address Allocation for Large Scale MANETs," *Proc. IEEE INFOCOM*, 2003, pp. 1304-1311.
- [6] Z. Yao and F. Dressler, "Dynamic Address Allocations for Management and Control in Wireless Sensor Networks," *Proc. HICSS*, 2007, pp. 292b.
- [7] C.E. Perkins and E.M. Rouyer, "Ad-hoc On-Demand Distance Vector Routing," *Proc. IEEE WMCSA*, 1999, pp. 90-100.
- [8] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad hoc Networks*, vol. 3, no. 3, May 2005, pp. 325-349.
- [9] J.N. Al-Karaki and A.E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Commun.*, vol. 11, no. 6, Dec. 2004, pp. 6-28.
- [10] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequence-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM*, 1994, pp. 234-244.
- [11] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," *IEEE Trans. Mobile Computing*, vol. 3, no. 3, July 2004, pp. 272-285.
- [12] K. Sohrabi et al., "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Pers. Commun.*, vol. 7, no. 5, Oct. 2000, pp. 16-27.
- [13] L.H. Yen and W.T. Tsai, "Flexible Address Configurations for Tree-Based ZigBee/IEEE 802.15.4 Wireless Networks," *Proc. AINA*, 2008, pp. 395-402.
- [14] H.I. Jeon and Y. Kim, "Efficient, Real-Time Short Address Allocations for USN Devices Using LAA (Last Address Assigned) Algorithm," *Proc. ICACT*, 2007, pp. 689-693.





**Sungjin Park** received the BS degree in computer engineering from Sejong University, Republic of Korea, in 2007. He is currently a master student with the Department of Information and Communication Engineering of Sejong University. His current research interests include embedded network systems.



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



**Eun Ju Lee** received the BS and MS degrees in computer engineering from Chungnam National University, Korea, in 1992 and 1994, respectively. Since 1992, she has been with ETRI, Daejeon, Korea. She is currently the senior member of engineering staff. Her current research interests include wireless sensor

networks and wireless communication and applications.



**Jae Hong Ryu** received the BS and MS degrees in electronic engineering from Chungbuk National University, Korea, in 1990 and 2005, respectively. Since 1991, he has been with LGIC and ETRI Korea. He is currently the senior member of engineering staff. His current research interests include embedded software,

wireless sensor networks, and telematics.



**Seong-Soon Joo** received his BS from Hanyang University in 1980, and received his MS and PhD degrees from the Seoul National University, Korea, in 1982 and 1989, respectively, all in electrical engineering. He joined ETRI in 1983, where he is the principal researcher on ubiquitous sensor networks. From

1996 to 1997, he was a visiting researcher with the Computer Science and Engineering Department of Arizona State University, USA. Since September 2004, he has been an adjunct professor with the Department of Broadband Network Engineering, University of Science and Technology, Daejeon, Korea. His research interests include soft computing, active networks, intelligent control for communication networks, and network architecture for the future Internet.