Tree-Based Data Broadcast in IEEE 802.15.4 and ZigBee Networks

Gang Ding, Zafer Sahinoglu, Philip Orlik, Jinyun Zhang, Bharat Bhargava

TR2006-089   December 2006

Abstract

This paper studies efficient and simple data broadcast in IEEE 802.15.4 based ad hoc networks (e.g. ZigBee). Since finding the minimum number of rebroadcast nodes in general ad hoc networks is NP-hard, current broadcast protocols either employ heuristic algorithms or assume extra knowledge such as position or two-hop neighbor table. However, the ZigBee network is characterized as low data rate and low cost. It cannot provide position or two-hop neighbor information, but it still requires an efficient broadcast algorithm that can reduce the number of rebroadcast nodes with limited computation complexity and storage space. To this end, this paper proposes self-pruning and forward node selection algorithms that exploit the hierachical address space in ZigBee networks. Only one-hop neighbor information is needed; a partial list of two-hop neighbors is derived without exchanging messages between neighboring nodes. The ZigBee forward node selection algorithm finds the minimum rebroadcast nodes set with polynomial computation time and memory space. Using the proposed localized algorithm, it is proven that the entire network is covered. Simulations are conducted to evaluate the performance improvement in terms of the number of rebroadcast nodes, number of duplicated receivings, coverage time, and communication overhead.

IEEE Commun Letter 2006

Tree-Based Data Broadcast in IEEE 802.15.4 and ZigBee Networks[1]

Gang Ding (corresponding author): School of Electrical and Computer Engineering, Purdue University, W. Lafayette, IN 47907, Phone: 765-409-4311, Fax: 765-494-0739, Email: dingg@ecn.purdue.edu

Zafer Sahinoglu: Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, Phone: 617-621-7588, Fax: 617-621-7550, Email: zafer@merl.com

Philip Orlik: Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, Phone: 617-621-7570, Fax: 617-621-7550, Email: porlik@merl.com

Jinyun Zhang: Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139, Phone: 617-621-7595, Fax: 617-621-7550, Email: jzhang@merl.com

Bharat Bhargava: Department of Computer Sciences, Purdue University, W. Lafayette, IN 47907, Phone: 765-494-6013, Fax: 765-494-0739, Email: bb@cs.purdue.edu

Abstract

This paper studies efficient and simple data broadcast in IEEE 802.15.4 based ad hoc networks (e.g., ZigBee). Since finding the minimum number of rebroadcast nodes in general ad hoc networks is NP-hard, current broadcast protocols either employ heuristic algorithms or assume extra knowledge such as position or two-hop neighbor table. However the ZigBee network is characterized as low data rate and low cost. It cannot provide position or two-hop neighbor information, but it still requires an efficient broadcast algorithm that can reduce the number of rebroadcast nodes with limited computation complexity and storage space. To this end, this paper proposes self-pruning and forward node selection algorithms that exploit the hierarchical address space in ZigBee networks. Only one-hop neighbor information is needed; a partial list of two-hop neighbors is derived without exchanging messages between neighboring nodes. The ZigBee forward node selection algorithm finds the minimum rebroadcast nodes set with polynomial computation time and memory space. Using the proposed localized algorithms, it is proven that the entire network is covered. Simulations are conducted to evaluate the performance improvement in terms of the number of rebroadcast nodes, number of duplicated receivings, coverage time, and communication overhead.

Index terms: broadcast, IEEE 802.15.4, ZigBee, ad hoc network

I. INTRODUCTION

The IEEE 802.15.4 standard [1] was approved in 2003 as a multiple access control (MAC) and physical (PHY) layer standard for low data rate, low power and low cost wireless personal area networks (WPANs). *The ZigBee*[2] *Alliance* [2] is a rapidly growing association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked monitoring and control applications. The ZigBee specifications were ratified in December 2004. And the ZigBee network specification [3] is one of the first standards for ad hoc and sensor networks.

In contrast to the intensive industrial activities on IEEE 802.15.4 and ZigBee, academic research on this subject is still at its early stage [4, 5], partly because the ZigBee specifications were not publicly available until June 2005 [2]. This paper addresses one of the most important research topics on ZigBee networks[3]: data broadcast. Due to the broadcast nature of wireless channels, a single transmission from a node is heard by all nodes within its transmission range. A simple broadcast approach is to flood the whole network by rebroadcasting at every node. But this approach consumes too much communication bandwidth, causing *broadcast storm problem* [6]. Some local network topology information should be employed to reduce the broadcast redundancy. Most current approaches assume that the transmission power can be controlled, and that the position or 2-hop neighbor information is easily available. However, these assumptions are not practical for ZigBee

[2] *ZigBee* takes its name from the zigzag flying of bees that forms a mesh network among flowers. It is an individually simple organism that works together to tackle complex tasks.

[3] In this paper, we will use *ZigBee network* to represent a wireless network that involves both IEEE 802.15.4 for physical and MAC layers and ZigBee protocols for network and higher layers.

networks because 1) A ZigBee device has limited computation and storage capacity. All the functionalities in MAC and upper layers are expected to be implemented in a single chip microcomputer, such as a 16-bit M16C [7]. So it can not afford to conduct sophisticated algorithms based on data structures that take a large memory space. 2) A ZigBee device should be of small size and low cost, so it cannot obtain accurate position information using extra hardware like GPS. 3) ZigBee networks are targeting low data rate and low power applications, so they cannot provide large communication bandwidth and power for exchanging position or neighbor information among neighbors. Even if such information can be obtained by a ZigBee device, it may not even have enough memory space to store it when the network size is large.

Given the above limitations, we are motivated to find a localized and light-weight broadcast algorithm for ZigBee networks. In a typical ZigBee network, the network addresses are organized in a hierarchical manner so that one node can easily identify addresses of its *tree neighbors*, including its parent and children. In this work, we exploit the hierarchical address assignment in ZigBee networks and propose two localized broadcast algorithms: self-pruning algorithm and forward node selection algorithm. Both algorithms select a subset of network nodes to rebroadcast, while every node in the network is still able to receive the packet. In the first algorithm, a node decides by itself whether it should rebroadcast or not; it need not rebroadcast if a certain subset of its 1-hop neighbors have already received the packet. In the latter algorithm, a node selects a subset of its 1-hop neighbors, called *forward nodes*, for forwarding so as to cover all its 2-hop tree neighbors. It is proven that the proposed algorithm is of polynomial computation complexity and memory complexity, and it provides an optimal solution of selecting the minimum number of forward nodes. In contrast, for general ad hoc networks, this optimum problem is NP-hard so that it can not be solved by any known algorithms of polynomial complexity.

This paper is organized as follows. Section II introduces IEEE 802.15.4 and ZigBee, and summarizes related work on broadcasting in general ad hoc networks. Assumptions and notations are given in Section III. The ZigBee on-tree self-pruning and on-tree forward node selection broadcast algorithms are presented in Section IV and Section V, respectively. Their performance is evaluated in Section VI. Section VII concludes the paper.

II. Background And Related Work

A. IEEE 802.15.4 and ZigBee

The IEEE 802.15.4 standard is for low-rate WPANs that require low power, low cost, and low complexity. An IEEE 802.15.4 device can be deployed in sensor networks, home automation, and industrial automation and monitoring etc. At the physical layer, IEEE 802.15.4 defines 27 channels of data rates 250 kb/s, 40 kb/s and 20 kb/s. At the MAC layer, IEEE 802.15.4 controls access to the radio channel using the Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) mechanism. IEEE 802.15.4 defines two types of devices: full function device (FFD) and reduced function device (RFD). An FFD can serve as a coordinator or a regular device. It can communicate with any other device. An RFD is a simple device that associates and communicates only with an FFD.

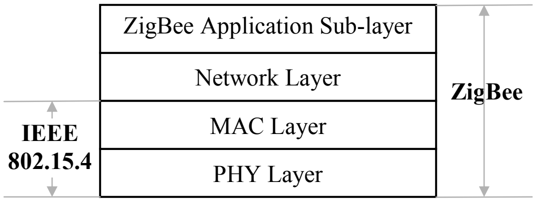


Figure 1. ZigBee protocol stack

Based on IEEE 802.15.4, the ZigBee Alliance specifies the standards for network and application sub-layer, as shown in Fig.1. The responsibilities of network layer [3] include joining/leaving a network, routing, security, discovering 1-hop neighbors and storing neighbor information. The ZigBee network layer assigns addresses and builds a hierarchical tree topology. A coordinator is responsible for starting a new ZigBee WPAN and setting network parameters such as the maximum allowable number of children n_m of each device and the maximum level d_m of the logical tree. The coordinator is the root of the tree with address 0. When a new device is willing to join a network, its MAC layer scans the available WPANs and notifies the network layer. After the upper layer selects a suitable WPAN, the network and MAC layer perform the association process with an existing device in the selected WPAN. If the existing device has enough address space, it will assign a free network address to the new device and make it one of its children. In case a child loses the association with its parent, it can initiate a rejoining process, called *orphaning*, and its parent will respond to resume the association. At any level d of the logical ZigBee tree, the network addresses are evenly separated by $Cskip(d) = (1 - n_m^{d_m - d})/(1 - n_m)$. The ZigBee routing algorithm is a hybrid of tree forwarding and ad hoc on-demand routing. Any node can route frames to its parent or direct children on the tree. For example, assuming a device at level d with address A receives a packet with destination address D, it will select one of its children with address $A + 1 + \lfloor (D - A - 1)/Cskip(d) \rfloor \cdot Cskip(d)$ [4] as the next hop if $A < D < A + Cskip(d-1)$; otherwise its parent is selected as the next hop. The RFDs are leaf nodes and can only forward the data packet to their own parents. If an FFD has enough computation capacity and storage space, it can be a router-capable FFD (RFFD) and store routes to devices other than its tree neighbors. An RFFD discovers a route by broadcasting a route request and waiting for replies from the destination or intermediate nodes, similar to the Ad hoc On-demand Distance Vector (AODV) routing protocol for general multi-hop ad hoc networks [8].

[4] $\lfloor x \rfloor$ represents the largest integer that is smaller than x.

B. Broadcast in Ad Hoc Networks

Most research on ad hoc networks models the network topology as a unit disk graph in which each node has a unit transmission range. Ni *et al.* [6] introduced the broadcast storm problem when every node rebroadcasts a packet once. To reduce the broadcast redundancy and avoid collisions during rebroadcasting, they introduced some simple algorithms. For example, the counter based algorithm rebroadcasts a packet only if the number of duplicated broadcast packets received during a waiting period is less than a threshold. The location based approach only rebroadcasts when the additional coverage by the rebroadcast is larger than a threshold. In [9], the authors improved the above algorithms by adaptively choosing the threshold as a function of the number of neighbors. Both simulation comparison [10] and theoretical analysis [11] have been conducted. These approaches are simple to implement, but they can not guarantee coverage of the whole network [10].

More complicated algorithms assume the knowledge of network topology in order to guarantee network coverage and reduce broadcast redundancy. When the global network information is available, the problem of selecting the minimum number of forward nodes is essentially the well-studied *set cover problem*, which is NP-hard [12]. But its solution can be approximated by a greedy algorithm [13] with an approximation factor of log(*n*), where *n* is the maximum number of neighbors. Since global network topology is not practically available, localized algorithms which only need the information of 1-hop and 2-hop neighbors are preferred. However, selecting a minimum subset of 1-hop neighbors to cover all 2-hop neighbors is still a set cover problem. For the special case when the network topology is a unit disk graph, the complexity of the corresponding forward node selection problem is not known, but it can be approximated by sophisticated algorithms with a constant factor [14]. Many heuristic algorithms have been proposed to decide if a node needs to rebroadcast. A Scalable Broadcast Algorithm (SBA) was proposed by Peng and Lu [15], which checks if a receiving node's 1-hop neighbors are all covered by previous transmissions from other nodes. If so, the receiving node need not rebroadcast the packet. Lim and Kim [12] and Stojmenovic *et al*. [16] proposed independently the same approach which is called *self pruning* and *neighbor elimination*, respectively. This paper will use *self pruning* to represent this idea. Peng and Lu [17] also proposed an efficient Ad Hoc Broadcast Protocol (AHBP), which greedily selects forward nodes from a node's 1-hop neighbors to cover all its 2-hop neighbors; all previously selected nodes on the same route need not be selected again and the node with the most uncovered neighbors is first selected. Other algorithms based on the similar greedy forward node selection idea include the *dominant pruning* [12] and the *multipoint relaying* [18]. This paper will use *forward node selection* to represent this idea. The size of the candidate forward node set can be further reduced by neighbor designation approaches in [19]. These localized forward node selection algorithms guarantee the coverage of the whole network, but informing 1-hop neighbors that who will be a forward node introduces a communication overhead. A detailed overview of broadcasting in ad hoc networks can be found in [20].

Research on energy efficient broadcast protocols is usually coupled with topology control, which tries to maintain a connected network topology with the minimum total transmission power [21]. But finding a minimum energy broadcast tree is NP-hard [22]. Many heuristic protocols have been introduced [23], which can approximate the optimal solution with a constant factor when the global network information is available. Some localized protocols have recently been proposed [24], but they need location or distance information and introduce extra communication overhead to exchange such information among nodes.

Due to the resource constraints in IEEE 802.15.4 and ZigBee networks, we make following assumptions in this paper:

- The distance between nodes and the position of nodes are not available.

- The transmission power of nodes is fixed and the same.
- The network topology is not necessarily modeled as a unit disk graph. But the symmetry of neighborhood is assumed. That is, if *A* is a neighbor of *B*, then *B* is also a neighbor of *A*.
- Addresses are hierarchically assigned. Hence, given the network address of a device, the addresses of its parent and children can be derived without information exchange.
- Every device maintains a table of only 1-hop neighbors. Each neighbor table entry consists of a neighbor's network address and the number of its children.

To facilitate the description, some notations are listed in Table 1, where *A* and *B* represent a set of nodes or a single node.

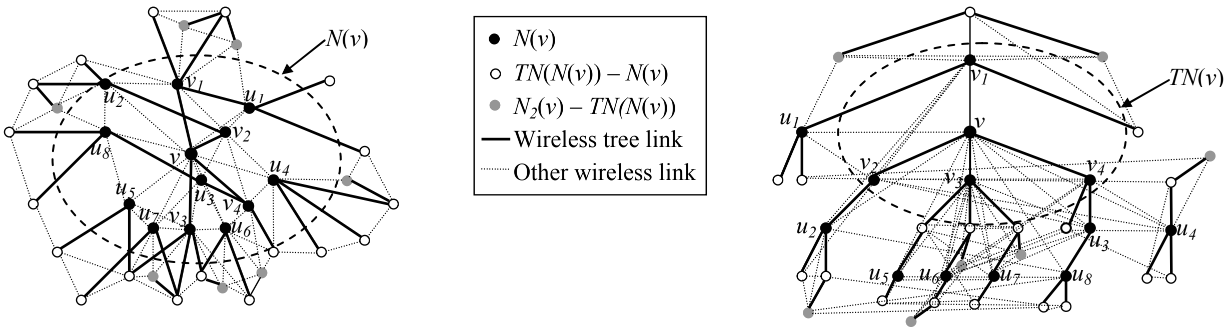
Table 1. Notations

$N_0(A)$ or $TN_0(A)$	A
$N_1(A)$ or $N(A)$	1-hop neighbors of A, including A
$N_k(A)$ $(k \geq 2)$	$N(N_{k-1}(A))$, k-hop neighbors of A
A covers B	$B \subseteq N(A)$
$TN_1(A)$ or $TN(A)$	1-hop tree neighbors of A, including A
$TN_k(A)$ $(k \geq 2)$	$TN(TN_{k-1}(A))$
A on-tree covers B	$B \subseteq TN(A)$
$A - B$	$A \cap \overline{B}$
$S(A)$	Candidate forward nodes of A
$C(A)$	To-be-covered nodes of A
$F(A)$	Forward nodes of A

III.ZIGBEE ON-TREE SELF-PRUNING ALGORITHMS

An on-tree self-pruning broadcast algorithm for ZigBee networks is presented in this section. Upon reception of a broadcast packet, a node decides whether to rebroadcast or not. Basically, after a source broadcasts a packet, all its 1-hop neighbors receive it. If they all rebroadcast the packet at the same time, catastrophic packet collisions may happen and delay the whole process of broadcast. To avoid collisions, every forward node waits for a random period of time before rebroadcasting. During this waiting period, a node *v* may receive the duplicated broadcast packet from another node *u*. So node *v* only needs to cover $N(v) - N(u)$, provided v knows 1-hop neighbors $N(u)$ of node u. If node v learns that all its 1-hop neighbors have already been covered before time out, it does not need to rebroadcast.

For a general ad hoc network, one issue with the above self-pruning algorithm is that the 2-hop neighbor information is assumed available to node v. When this assumption does not hold, node v can only know that the source node u of a duplicate packet has been covered so that v still need to cover $N(v) - u$. Node v can be self-pruned only if it has received the broadcast packet from all its 1-hop neighbors in $N(v)$, which does not happen with a high probability during a short waiting period, especially when $N(v)$ is very large. As a result, the self-pruning broadcast algorithm would perform poorly when applied to ZigBee networks where the 2-hop neighbor information is not available. On the other hand, by exploiting the tree structure of ZigBee address space, a node can find addresses of a partial list of 2-hop neighbors without introducing any communication or storage overhead. In other words, given the address of a 1-hop neighbor in $N(v)$ and its number of children, one can determine the addresses of all its tree neighbors in $TN(N(v))$.



a. Physical topology **b. Logical ZigBee tree topology**

Figure 2. A ZigBee network topology

Fig. 2 displays a ZigBee network when $n_m = 3$, including both physical topology and logical tree topology. For node v at the center, all its 1-hop neighbors $N(v)$ (black dots) are located within its transmission range as shown in Fig. 2a. Among these 1-hop neighbors, four tree neighbors belong to $TN(v)$: a parent v_1 and three children v_2, v_3 and v_4. The other eight 1-hop neighbors, u_1 to u_8, can be located anywhere on the logical ZigBee tree in Fig. 2b. Assuming the address of node v is A_v, the depth of node v in the logical ZigBee tree is d, and it is the kth child of its parent v_1, then the address of its parent would be $A_v - (k-1)Cskip(d) - 1$ and the address of its ith $(1 \le i \le n_m)$ child is $A_v + 1 + (i-1)Cskip(d+1)$. More details of address assignment in ZigBee networks can be found in [2]. Similarly, given the network addresses of v_1 to v_4 and u_1 to u_8, node v can further identify their parent and children, which form $TN(N(v))$ represented by white dots in Fig. 2. For any 1-hop neighbor of node v, for example node v_1, it may have its own 1-hop neighbors other than its tree neighbors, which are not known by node v. Therefore, $TN(N(v))$ is only a subset of $N_2(v)$. The gray dots in Fig. 2 represent all nodes in $N_2(v) - TN(N(v))$. The tree neighbors are connected by solid lines in Fig. 2, while any two nodes that are not tree neighbors but are within transmission range of each other are connected by dotted lines. According to the self-pruning algorithm for general ad hoc networks, node v in Fig. 2 needs to receive the duplicated broadcast packet from all its twelve 1-hop neighbors before it can prune itself; whereas with the on-tree self-pruning algorithm we propose, node v can prune itself once it learns that its four tree neighbors, v_1 to v_4, have been already covered. The coverage information can be acquired by receiving the duplicated broadcast packet from either v_1 to v_4, or their tree neighbors.

Fig. 3 gives the On-tree Self-pruning Rebroadcast (OSR) algorithm when a node v receives a broadcast packet from node u. The localized OSR algorithm actually guarantees that the whole network will be covered, as proven by Theorem 1.

Theorem 1. *When every node in a ZigBee network runs the OSR algorithm, a broadcast packet from any source can reach all the other nodes, provided the network is physically connected.*

Proof: See Appendix A.1.

```
On-tree Self-pruning Rebroadcast (OSR)
Input: receiving node $v$ and broadcasting node $u$
Output: none
-------------------------------------------------------
if it is the first time to receive a broadcast packet,
    Set an on-tree to-be-covered set: $TS = TN(v) - TN(u)$
    if $TS = Ø$,
        Drop the packet.
    else
        Buffer the packet and record its broadcast ID.
        Start a timer with random timeout.
    end if
else
    if the early copy of this packet is waiting,
        Update $TS = TS - TN(u)$
        if $TS = Ø$,
            Drop the packet and stop the timer.
        end if
    else
        Drop the duplicated broadcast packet.
    end if
end if

if the timer expires
    broadcast the packet.
end if
```

Figure 3. ZigBee on-tree self-pruning rebroadcast algorithm

IV. ZigBee On-Tree Selection Algorithm

This section introduces an optimal solution to forward node selection problem in ZigBee networks. After a source initiates a broadcast process, the forward node selection algorithm will be executed at every rebroadcasting node to further selects a subset of its 1-hop neighbors as the new forward nodes. In a ZigBee network, as indicated in the previous section, although the complete list of 2-hop neighbors $N_2(v)$ is not available, a partial 2-hop neighbor list $TN(N(v))$ can be easily found without any communication overhead. So the *forward node selection problem for ZigBee networks* reduces to

Find the smallest forward node set $F(v) \subseteq N(v)$ to on-tree cover $TN(N(v))$.

```
Optimal On-tree Selection (OOS)
Input: $N(v)$ and broadcasting node $u$ with $F(u)$
Output: $F(v)$
1. if $v$ is the source of broadcast
        $S(v) = N(v) - v,$                                        (1)
        $C(v) = TN(N(v)) - N(v).$                                 (2)
   else /* $v$ must receive the broadcast packet from another node $u$. */
        $S(v) = N(v) - v - TN(u) - F(u),$                         (3)
        $C(v) = TN(N(v)) - N(v) - TN_2(u) - TN(F(u)).$            (4)
   end if
   Initialize $F(v) = \emptyset$.
2. Construct a forest composed of all nodes in $S(v) \cup C(v)$, which is part of the
logical ZigBee tree of the whole network.
3. for every tree in the forest,
      for every level in the tree, starting from the bottom,
         for every node $w \in C(v)$, from left to right,
            if $x$ = Parent($w$) $\in S(v)$,
                $F(v) = F(v) \cup \{x\}$, $S(v) = S(v) - \{x\}$, $C(v) = C(v) - TN(x)$.
            else there must exist $y \in S(v)$ such that $w$ = Parent($y$)
                $F(v) = F(v) \cup \{y\}$, $S(v) = S(v) - \{y\}$, $C(v) = C(v) - \{w\}$.
            end if
         end for
      end for
   end for
```

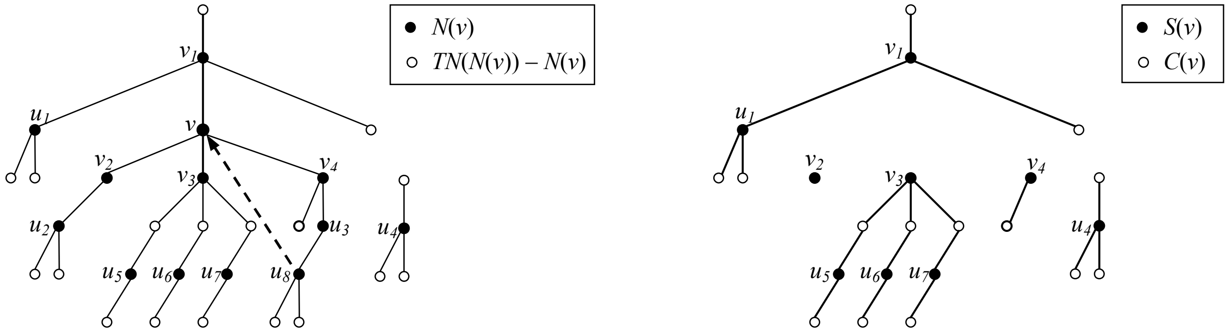
Figure 4. Optimal on-tree forward node selection algorithm

In this section, we first propose an Optimal On-tree forward node Selection algorithm (OOS) that solves the problem in polynomial time. It is further refined to the ZigBee On-tree Selection algorithm (ZOS) that solves the problem with polynomial memory space. It is proven that the proposed algorithm guarantees to cover the whole network.

A. Optimal On-tree Selection Algorithm

Given a node *v* that can be the source of a broadcast packet or a selected forward node receiving packet from node *u*, the localized algorithm OOS given in Fig. 4 selects the minimum size forward node set *F*(*v*) from *N*(*v*) to on-tree cover *TN*(*N*(*v*)).

The first step in the OOS reduces the size of the candidate forward node set *S* and the to-be-covered set *C*. If node *v* is the source or a forward node, it will definitely rebroadcast and cover all its 1-hop neighbors, so it does not need to be selected as a forward node again, and all its neighbors do not need to be covered again, as shown in (1) and (2). If node *v* receives the broadcast packet from another forward node *u*, it can remove set $TN(u)$ and $TN_2(u)$ from *S* and *C*, respectively, because *u* would have already selected its own forward nodes from $N(u) \supseteq TN(u)$, which will on-tree cover $TN(N(u)) \supseteq TN_2(u)$. Similarly, since *v* knows $F(u)$, it can further remove $F(u)$ and $TN(F(u))$ from *S* and *C*, respectively, because these forward nodes are to rebroadcast and on-tree cover their own tree neighbors. As an example based on the ZigBee network in Fig. 2, Fig. 5a shows the logical ZigBee tree known by node *v* when it receives a broadcast packet from node u_8, along with $F(u_8) = \{v, u_2\}$. Node *v* can find sets *S* and *C* using (3) and (4). In step 2 of OOS algorithm, the trees involving *S* and *C* are constructed as shown in Fig. 5b.



a. Local ZigBee tree topology known by node *v*

b. Step 2 in OOS algorithm

Figure 5. An example of using OOS algorithm

In step 3, a minimum size forward node set $F(v)$ is selected from set *S* in order to on-tree cover set *C*. For every tree constructed in step 2, the OOS algorithm goes through it level by level in a top-to-bottom direction and from left to right at each level. For example, in Fig. 5b, for the tree rooted at node v_3, nodes u_5, u_6 and u_7 are selected in order to on-tree cover their children in *C*, respectively, and v_3 need not be selected because all its children have already been covered by nodes u_5, u_6 and u_7. In contrast, if a greedy algorithm as in [17] is applied to first select the node with the maximum number of uncovered neighbors. v_3 will be first selected, so will u_5, u_6 and u_7, which is not optimal. In the OOS, the selected minimum size forward node set of node v becomes $F(v) = \{v_1, v_4, u_1, u_4, u_5, u_6$ and $u_7\}$.

The correctness and optimality of the OOS algorithm is given in Theorem 2.

Theorem 2*. For a node v, the OOS algorithm finds a minimum size forward node set from N(v) to on-tree cover TN(N(v)). The computational complexity is* $O(nn_m log(n))$.

Proof: See Appendix A.2.

B. ZigBee On-tree Selection Algorithm

The OOS algorithm assumes that the construction and traversing of a tree does not introduce any extra cost. When implementing the algorithm in the real system, however, this assumption is not practical because a ZigBee device usually has very limited memory and computation capacity. Since a large part of the memory has already been used by the routing table and neighbor table, it may not even have enough space to store the whole forest or even a tree of $O(nn_m)$ nodes. It needs considerable computation capacity to generate, merge and delete nodes on the tree. Even if a tree is constructed, it requires extra effort to sort and traverse it level by level.

A more practical algorithm, ZigBee On-tree Selection (ZOS) algorithm, is proposed in Fig. 6. It only needs $O(n)$ of memory space while keeping the same computational complexity as the OOS. The ZOS has the same result as the OOS, but it is implemented in a more memory-efficient way. Basically, the OOS works on both S and C with size $O(n)$ and $O(nn_m)$, respectively, while the ZOS only works on S which is stored in the existing neighbor table. Since C just contains the tree neighbors of set S, it can be easily derived from S whenever needed. A small memory of size M is required to temporarily store nodes at a level in C. The ZOS algorithm is explained as follows.

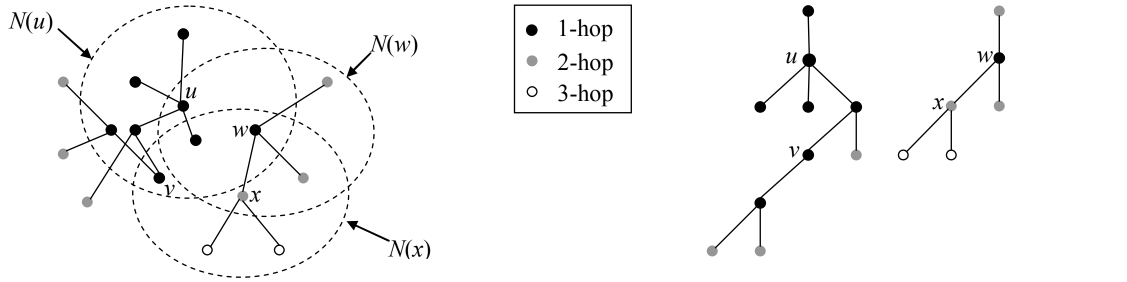
```
ZigBee On-tree Selection (ZOS)
Input: $N(v)$ and broadcasting node $u$ with $F(u)$
Output: $F(v)$
1. Same as step 1 in OOS algorithm.
2. Given $N(v)$ stored in a neighbor table, sort $N(v)$ by level and network address at each level.
   Initialize a piece of memory $M$ of size $n_l$ equal to the maximum size of all levels.
   for every node $x$ in $N(v)$
       if $x \in S$
           Status($x$) = "Initial"
       else
           Status($x$) = "alreadyForward"
       end if
   end for
   currentLevel = the depth of bottom level.
3. for (each node $x$ in currentLevel) and (Status($x$) = "Initial"),
       if ($x$ does not have any children) or ($x$'s every child is in (currentLevel + 1) or $M$)
          Status($x$) = "nonForward"
       else
          Status($x$) = "Forward"
       end if
   end for
   Empty $M$.
4. for (every node $y$ in (currentLevel + 1)) and (Status($y$) ≠ "alreadyForward"),
       if $w$ = Parent($y$) is not in currentLevel
           Add $w$ to $M$
           if (Status($y$) = "nonForward") and ($x$ = Parent($w$) belongs to (currentLevel – 1)),
                Status($x$) = "Forward",
           else
                Status($y$) = "Forward",
           end if
       end if
   end for
5. currentLevel = currentLevel – 1,
   if currentLevel = 0,
       Go through the whole neighbor table and collect all nodes with status "Forward",
       exit
   else
       go to 3.
   end if
```

Figure 6. ZigBee on-tree forward node selection algorithm

In step 2 of the ZOS algorithm, every node in S is a candidate forward node with status "*Initial*", while any node in $N(v) – S$ is marked as "*alreadyForward*" because the ZOS does not select forward node from them and has no concern of covering their tree neighbors. Starting from the bottom level and from left to right at each level, step 3 checks if every node *x* in *S* at *currentLevel* needs to be selected to on-tree cover all its children in the immediate lower level. If all of node *x*'s children exist in *currentLevel* + *1* of the neighbor table or in *M* which stores nodes in *C* that have already been covered by some previously selected forward nodes, *x* does not need to on-tree cover its children and is assigned status "*nonForward*". Step 4 further makes sure that the parent *w* of one node *y* is covered by either *y*'s grandparent *x* or *y* itself, as explained in the proof of Theorem 2 (see Fig. A.1). The covered parent is copied to *M* for the use of checking upper levels in step 3. Basically the ZOS algorithm is doing the same task as the OOS algorithm, so it obtains the same result as OOS, but it only uses neighbor table and a small piece of memory *M*. It is evident that, at any time, *M* only needs to save parents of at most one level of nodes in the neighbor table. Hence its size can be at most n_l = O(n). Every node with status "*alreadyForward*" never changes the status, but the nodes with status "*Initial*" switch to either "*Forward*" or "*nonForward*" in step 3, depending on if their children need to be covered. Step 4 may further change a node's status from "*nonForward*" to "*Forward*" when its parent must be covered by it. For the computational complexity, sorting the neighbor table in step 2 can be achieved by calculating the depth of every node, ordering levels from bottom to up, and sorting nodes by their network addresses at each level. The complexity of step 2 is O($n$log(n_l)). The step 3 and 4 need to search every neighbor's children at one level, with complexity O(nn_mlog(n_l)). Step 5 only takes O(n). So the total computational complexity is O(nn_mlog(n_l)).

C. Network Coverage of ZOS Algorithm

The above ZOS algorithm is used by every forward node to select its own forward nodes locally. After a node rebroadcasts the packet, only its selected forward nodes will rebroadcast. Yet it is not proven if the network will be covered globally. When applying the localized forward node selection algorithm, a non-forward node may be selected as a forward node later. As shown in Fig. 7, an initially non-forward node *v* for node *u* is selected as a forward node later by node *x*. It is proven by theorem 3 below that node *v* need not rebroadcast for node *x* and the whole network will eventually be covered.



a. Physical topology **b. Logical ZigBee tree topology**

Figure 7. An example that a non-forward node is selected as a forward node later

Theorem 3. *Given an arbitrary broadcast source, if the source and every forward node uses ZOS algorithm to select its own forward nodes and rebroadcasts the packet once, every non-forward node drops the packet and never rebroadcasts even if it is selected as a forward node later, the whole network is guaranteed to be covered and the broadcast process will automatically stop when all nodes are marked as either forward node or non-forward node. This process takes at most $2d_m$ steps.*

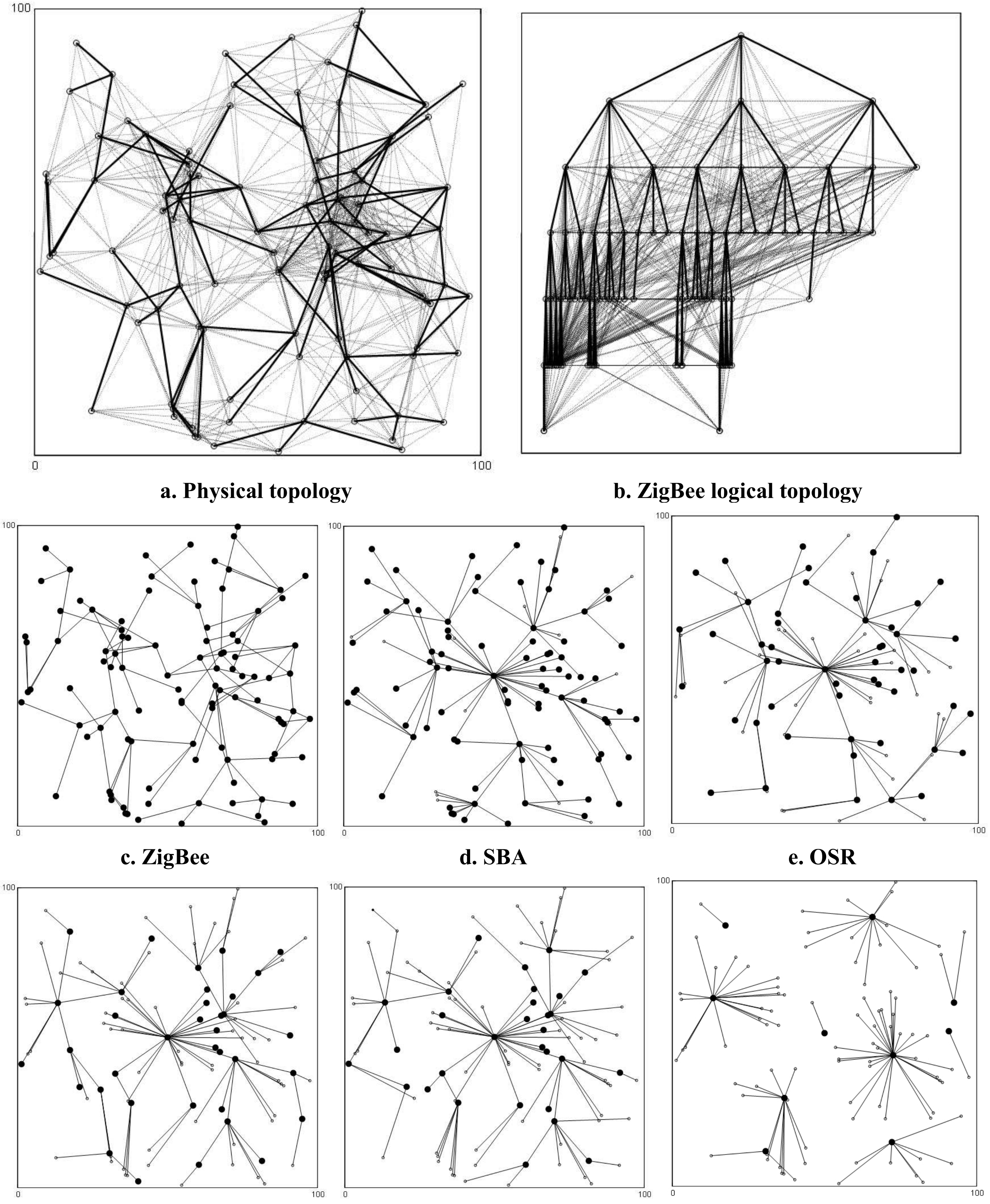
Proof: See Appendix A.3.

The self-pruning technique introduced in the previous section can be integrated into the above forward node selection algorithm, but a forward node needs to ensure that all its 1-hop neighbors, instead of 1-hop tree neighbors, have been covered before it stops rebroadcasting. We omit the proof in this paper.

V. PERFORMANCE EVALUATION

The proposed broadcast algorithms for ZigBee networks are evaluated by using an event-driven simulator we developed in MATLAB, which includes a ZigBee network generator. We assume an ideal case when there is no packet loss in MAC and physical layer. The reliability issue is recently studied in [25]. The network topology is generated within a 100 m × 100 m square area. The location of each node is randomly generated. In addition to the proposed OSR and ZOS algorithms, four other algorithms are simulated for comparison. 1) The SBA [15] in which a node is self-pruned only if all its physical 1-hop neighbors are covered; 2) The AHBP algorithm [17] in which a greedy algorithm is employed to select forward node from $N(v)$ to cover $TN(N(v))$; 3) The ZigBee broadcast algorithm currently used by the ZigBee specification, in which only tree neighbors rebroadcast; 4) The Global algorithm in which greedy algorithm is applied to the whole network, assuming the global network topology is known. Although the global information is not available for each ZigBee device, this Global algorithm gives an approximation of the global optimal solution of finding the minimum number of nodes to rebroadcast. In the greedy algorithm adopted by the above AHBP and Global algorithms, every node counts its neighbors that are not yet covered. The node with the maximum number of uncovered neighbors is selected first. Figure 8 illustrates a ZigBee network of 100 nodes and the rebroadcast nodes resulted from the above six algorithms. Both the physical topology and logical ZigBee topology are displayed, in which the wireless link between each pair of tree neighbors is represented by a solid line, while the other wireless links are represented by dash lines. The number of rebroadcast nodes is 100, 85, 51, 36, 31, and 10 for ZigBee, SBA, OSR, AHBP, ZOS, and Global algorithms, respectively, which is in decreasing order and reflects the increase of broadcast efficiency.

Fig. 9, 10, and 11 show the number of rebroadcast nodes, the average number of duplicated packets received by each node, and the time of covering the whole network, respectively. Each point on the figure represents the average of 100 simulation results, along with 95% confidence interval. Different randomly generated network topologies are used in each simulation run. The network size varies from 30 to 300 nodes. The maximum number of children n_m and the depth d_m are 3 and 6, respectively. The transmission range is a circle around each node of radius 25m. For the broadcast delay, we adopt a randomly generated waiting period between 0 and 1 millisecond for each broadcast, therefore the resultant coverage time is only used for comparison.



a. Physical topology

b. ZigBee logical topology

c. ZigBee

d. SBA

e. OSR

f. AHBP

g. ZOS

h. Global

Figure 8. ZigBee topology and rebroadcast nodes selected by six algorithms
Rebroadcast nodes are represented by big black dots in c to h. For algorithms ZigBee, SBA, OSR, AHBP, ZOS, and Global, the number of rebroadcast nodes is 100, 85, 51, 36, 31, and 10; the average number of duplicated packets is 17.6, 15.3, 9.42, 7.06, 6.39, and 1.89; the coverage time is 4.23 ms, 1.23 ms, 1.09 ms, 1.73 ms, 1.34 ms, and 0.954 ms.

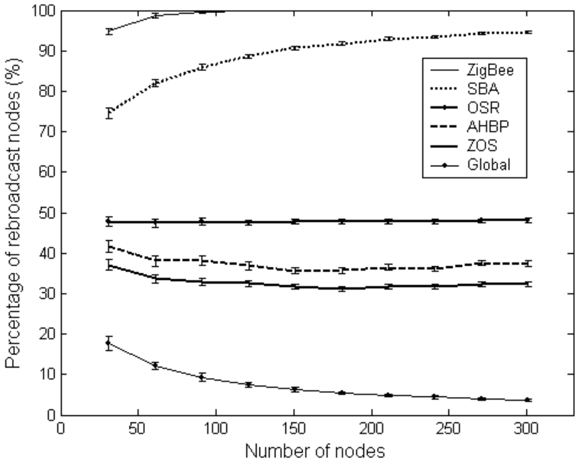


Figure 9. Number of rebroadcast nodes

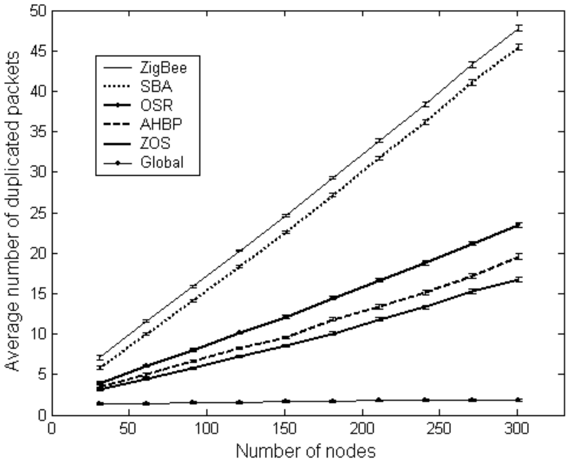


Figure 10. Average duplicated packets

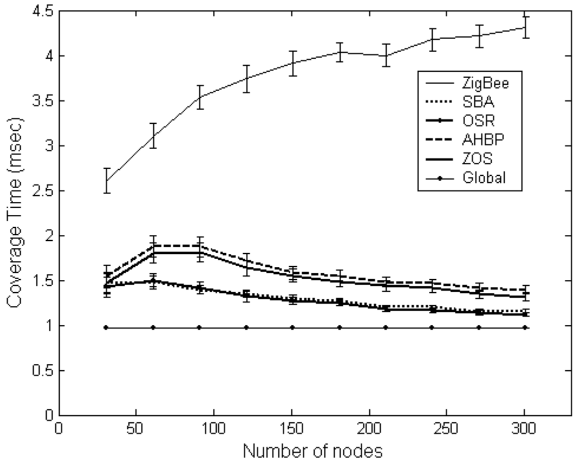


Figure 11. Coverage time

According to Fig. 9, the broadcast efficiency increases in the order of ZigBee, SBA, OSR, AHBP, ZOS, and Global. It can be concluded that 1) the flooding based ZigBee algorithms requires almost every node to rebroadcast; 2) the Global algorithm needs the least number of rebroadcast nodes due to the availability of global network information; 3) the forward node selection algorithms AHBP and ZOS selects less rebroadcast nodes than the self-pruning algorithms SBA and OSR; 4) the proposed OSR and ZOS algorithms outperform the SBA and AHBP algorithms, respectively. This agrees with the previous analysis that the OSR algorithm can do self-pruning when tree neighbors in *TN*(*v*) are covered while the SBA algorithm waits for all neighbors in *N*(*v*) to be covered before self-pruning; the

ZOS algorithm gives the optimal solution of selecting the minimum number of forward nodes while AHBP uses a greedy algorithm to select forward nodes.

For the number of duplicated packets received by each node shown in Fig. 10, similar conclusions to the number of rebroadcast nodes can be drawn. In contrast to less than two duplicated packets by the Global algorithm, all the other localized algorithms introduce significant broadcast redundancy, which increases with respect to the network size. Basically, the number of duplicated packets is determined by both the number of rebroadcast nodes and the number of neighbors covered by each broadcast. When the transmission range is fixed and the nodes are uniformly distributed, the number of duplicated packets is approximately proportional to the number of rebroadcast nodes.

For the coverage time displayed in Fig. 11, the ZigBee broadcast algorithm takes the longest time because only tree neighbors respond to each broadcast while all the other neighbors drop the packet even if they can successfully receive it. The Global algorithm offers the fastest coverage because all the globally selected forward nodes start broadcasting after a random waiting period and there is no further rebroadcasts. The self-pruning algorithms SBA and OSR cover the whole network a little faster than the forward node selection algorithms AHBP and ZOS. It is also observed that their coverage time converges to that of the Global algorithm when the network size increases.

A. Transmission Range

To evaluate the proposed broadcast algorithms in different scenarios, we change the transmission range and the network size. The transmission radius of 25m is used in the above simulation. This subsection gives the results when a small transmission radius 10m and a large transmission radius 55m are used, as shown in Fig. 12, 13, and 14. The relationship between different algorithms is the same as before. But the performance of some algorithms is more likely affected by the change of transmission range.

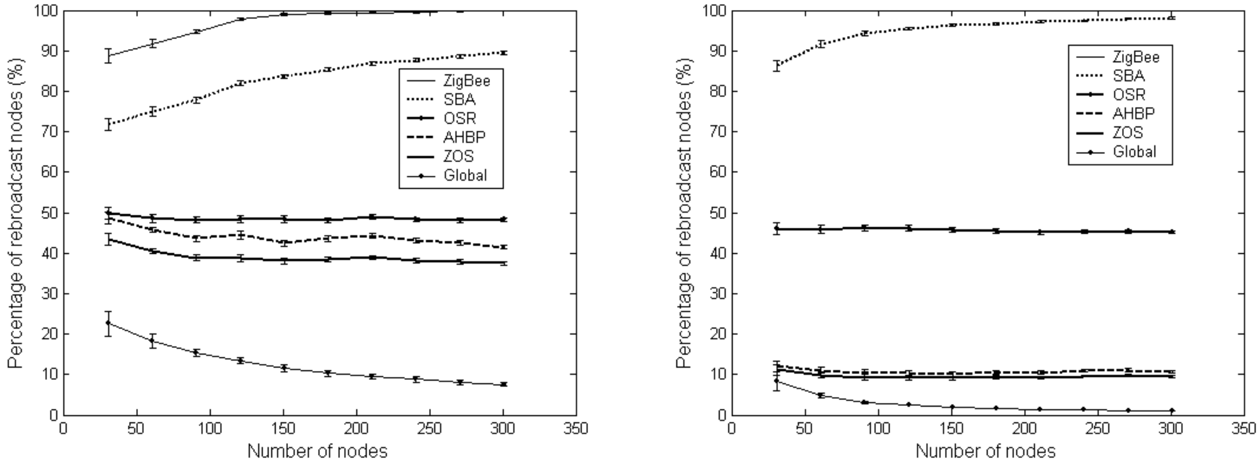


Figure 12. Number of rebroadcast nodes when transmission radius is 10m (left) and 55m (right)

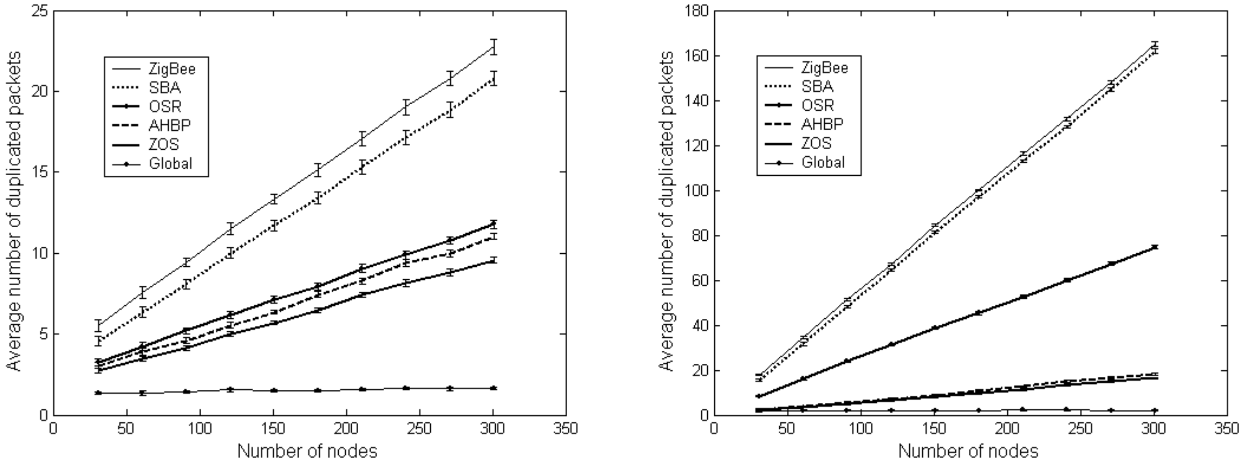


Figure 13. Average duplicated packets when transmission radius is 10m (left) and 55m (right)

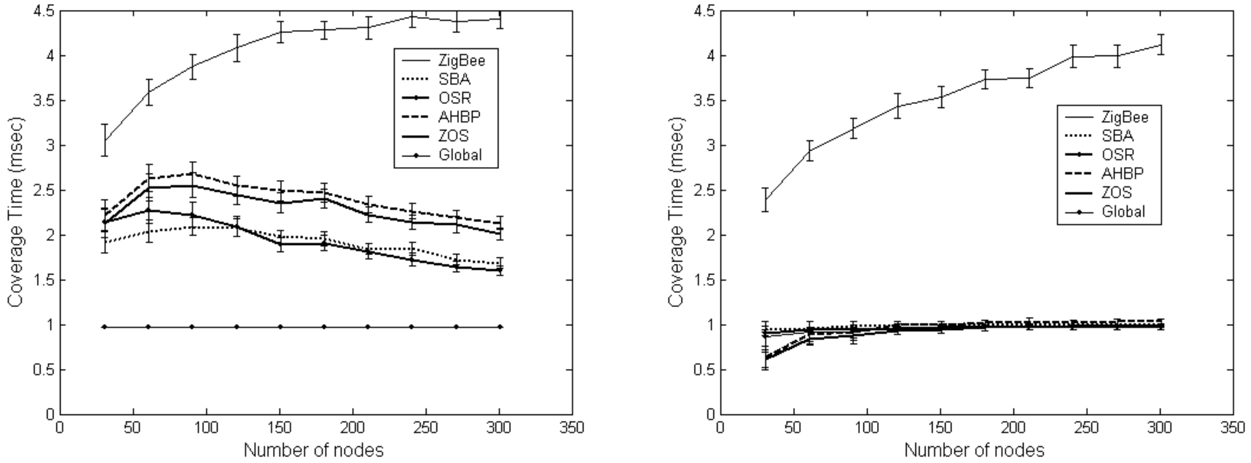


Figure 14. Coverage time when transmission radius is 10m (left) and 55m (right)

For the number of rebroadcast nodes shown in Fig. 9 and 12, it is observed that the performance of ZOS algorithm is significantly increased when the transmission radius increases, while the performance of OSR algorithm does not change much. This is because, when the transmission range increases, a node covers more neighbors and there is a better chance that all tree neighbors of a node already exist in the neighborhood. Hence this node does not need to be selected as a forward node by the ZOS algorithm. The OSR algorithm always waits for the coverage of all its tree neighbors, which is not significantly affected by the transmission range.

The coverage time shown in Fig. 11 and 14 decreases as the transmission range increases; because more neighbors can be covered in one transmission. In most cases, self-pruning algorithms SBA and OSR offer a faster coverage than the forward node selection algorithms AHBP and ZOS. This is due to the fact that in the forward node selection algorithms, nodes far away from the source in the logical topology can not receive the broadcast packet faster even if they become physically closer to the source when the transmission range increases. However, when the transmission range is increased to 55m, the forward node selection algorithms cover the whole network faster because the nodes are no longer far away from the source. It is also observed that the coverage time converges to that of the global algorithm when the transmission range increases.

B. Network Size

In the above scenarios, we tested ZigBee networks of up to 300 nodes. A ZigBee network can actually support more nodes. The maximum size of the network depends on the ZigBee network parameters n_m and d_m. In this subsection, small ZigBee networks of up to 85 nodes and large ZigBee networks of up to 1023 nodes are studied. The network parameters are given in Table 2. The simulation results are given in Fig. 15, 16, and 17.

Table 2. ZigBee network parameters

Network size	n_m	d_m	Simulated size
Small	4	3	5 – 85
Medium	3	6	30 – 300
Large	2	9	23 – 1023

The relationship between different algorithms is the same as before, but the impact of ZigBee network parameters on the algorithms is different. For the number of rebroadcast nodes shown in Fig. 9 and 15, the Global algorithm obviously performs the best and it scales well when the network size increases. The percentage of rebroadcast nodes using the self-pruning algorithms SBA and OSR increases when the ZigBee network size increases. But the forward node selection algorithms AHBP and ZOS are not significantly affected by the change of network size. There could be an implicit relationship between ZigBee network parameters n_m, d_m and the number of forward nodes. This needs to be justified when every possible combination of n_m and d_m is evaluated, which is one of our future research tasks.

For the coverage time shown in Fig. 11 and 17, it is interesting to note that, although there is an overshoot when the network size is small, the coverage time of algorithms SBA, OSR, AHBP, and ZOS all converges to that of the Global algorithm when the network size is increasing. This phenomenon is similar to the last scenario when the transmission range increases. It is because that, in the same area, when the total number of nodes increases, the density of neighborhood increases correspondingly so that a transmission will cover more nodes.

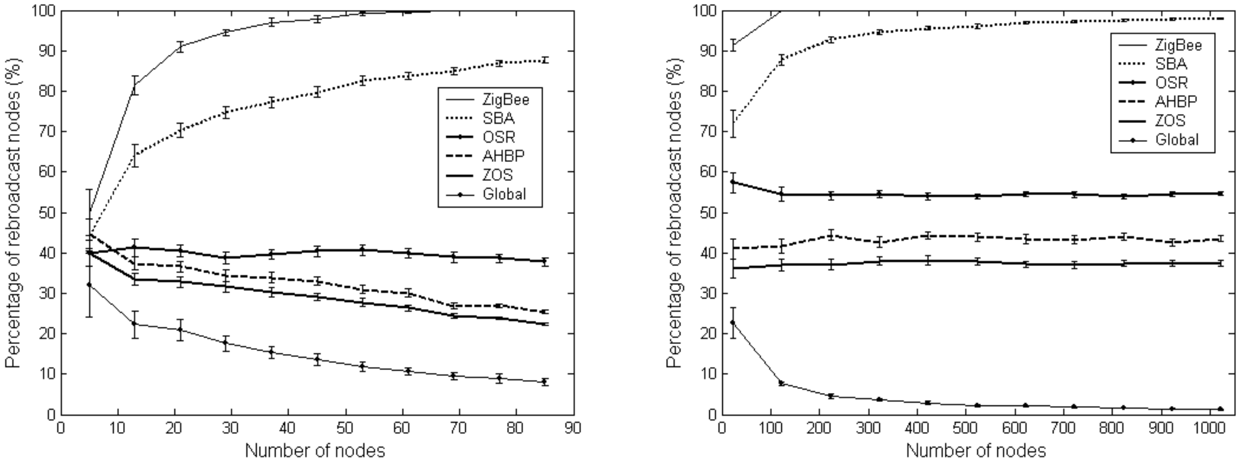


Figure 15. Number of rebroadcast nodes of small (left) and large (right) ZigBee networks

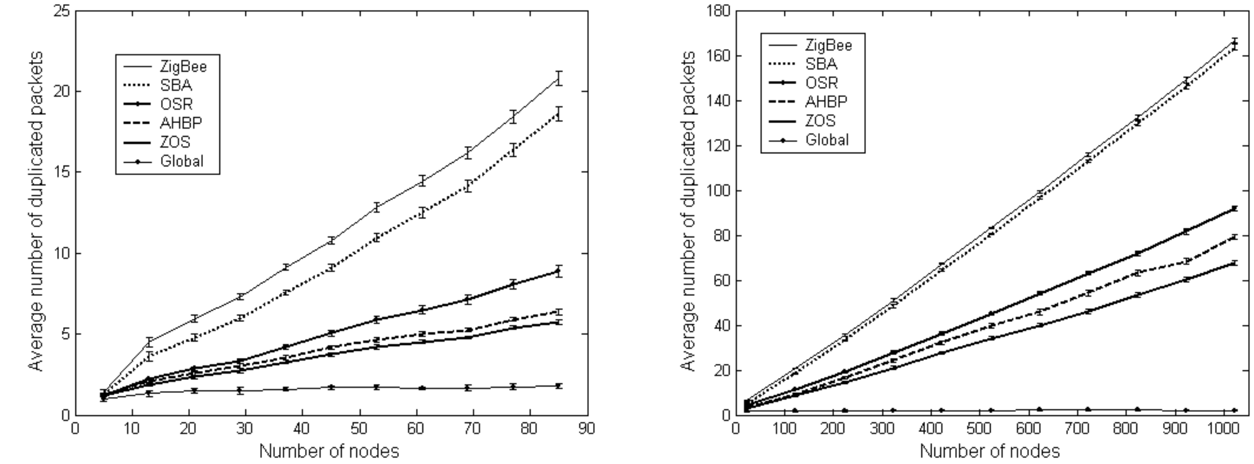


Figure 16. Average duplicated packets of small (left) and large (right) ZigBee networks

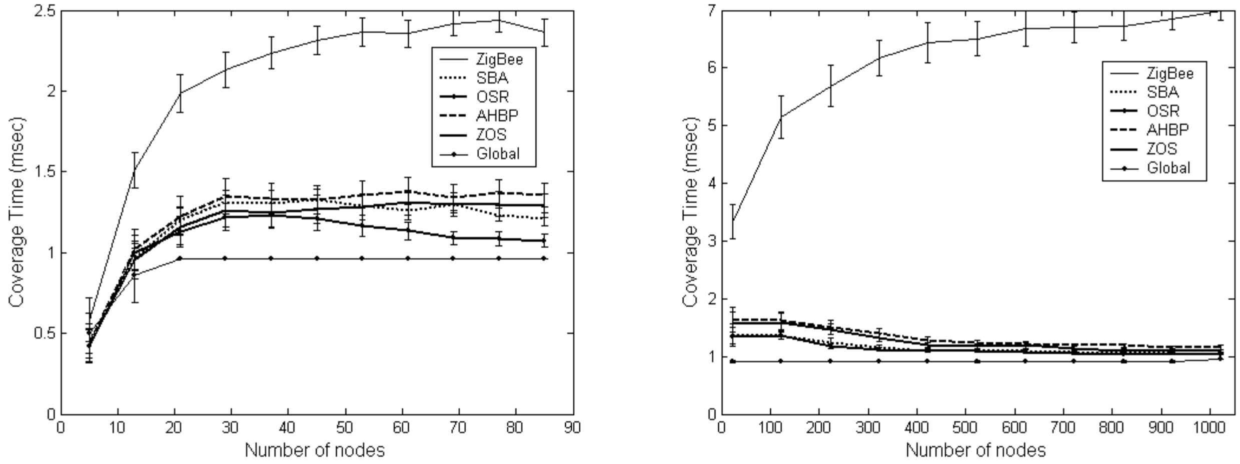
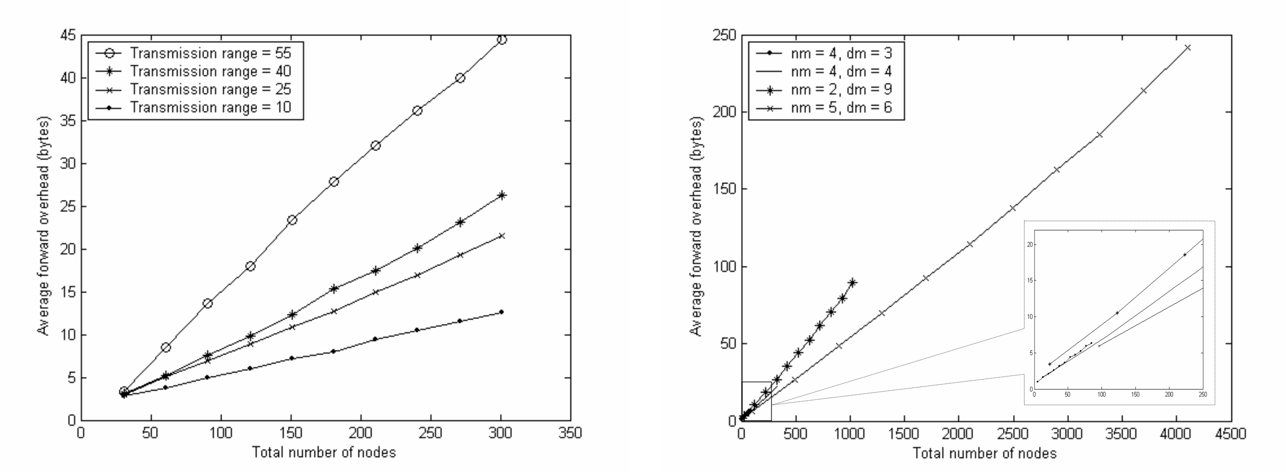


Figure 17. Coverage time of small (left) and large (right) ZigBee networks

C. Forward overhead

According to the above results, the proposed ZOS algorithm outperforms the OSR algorithm in general. However, it should be noted that the forward node selection algorithm introduces a communication overhead when a rebroadcasting node sends a list of forward nodes, along with the original broadcast packet, to all its neighbors. Assuming the network address of each selected forward node needs 2 bytes, Fig. 18 shows the average communication overhead of each forward node when the transmission range and the ZigBee network parameters vary. When the transmission range increases, every rebroadcast node will have more neighbors and more selected forward nodes, hence the communication overhead increases with respect to the transmission range. But it should be noted that the total communication overhead may not increase because the number of total forward nodes drops when the transmission range increases. For different network size, the communication overhead linearly increases when the network size increases. But it increases slower when the number of children n_m is larger.



a. Different transmission range **b. Different network size**

Figure 18 Communication overhead of ZOS algorithm

A final note about the communication overhead of forward node selection algorithms is that a node need not send the list of selected forward nodes every time it rebroadcasts. If a node *v* has sent the list once and its neighbor table has not been changed since then, *v* need not send the same list of forward nodes when it is rebroadcasting later, because every neighbor *u* can save the status of "*Forward*" or "*nonForward*" in its own neighbor table entry for node *v*. In case a new neighbor *u* does not know its status for node *v*, it just rebroadcasts the packet like a selected forward node for *v*.

VI. CONCLUSION

Given the fact that only physical 1-hop neighbors and logical tree neighbors are known in ZigBee networks, this paper has proposed two broadcast algorithms for IEEE 802.15.4 and ZigBee networks: on-tree self-pruning rebroadcast algorithm OSR and ZigBee on-tree selection algorithm ZOS. The ZOS algorithm selects less number of rebroadcast nodes than the OSR algorithm, but it introduces an extra communication cost to carry the list of selected forward nodes. The proposed OSR and ZOS algorithms outperform the flooding based ZigBee broadcast algorithm, and the self-pruning algorithm SBA and forward node selection algorithm AHBP for general ad hoc networks. The proposed broadcast algorithms are feasible in terms of both computation complexity and memory space complexity.

In the real ZigBee networks, in addition to minimizing the number of rebroadcast nodes, we need to consider the robustness of the broadcast algorithm, which means that the broadcast algorithm should allow some kind of redundancy in order to cover the whole network even if the 1-hop neighbor information is not up-to-date or the nodes are moving. The trade-off between broadcast efficiency and reliability is studied in [25]. Future works include investigating the relationship between system performance and the ZigBee network parameters, the impact of dynamic network topology and power constraints on the broadcast algorithm.

REFERENCES

[1] IEEE Standard 802, part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs), 2003.

[2] http://www.zigbee.org

[3] ZigBee Alliance, Network Specification, Version 1.0, December 2004.

[4] G. Lu, B. Krishnamachari, and C. Raghavendra, "Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks," In *Proceedings of the Workshop on Energy-Efficient Wireless Communications and Networks*, 2004.

[5] J. Zheng and M. Lee, "Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?: A Discussion on a Potential Low Power, Low Bit Rate Standard," *IEEE Communications Magazine*, 42(6): 140-146, 2004.

[6] S. Ni, Y. Tsing, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom)*, 1999.

[7] http://www.renesas.com

[8] C. E. Perkins and E.M. Royer, "Ad Hoc On-Demand Distance Vector Routing," In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, 90-100, 1999.

[9] Y. Tseng, S. Ni, and E. Shih, "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network," *IEEE Transactions on Computers*, 52(5): 545-557, 2003.

[10] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 194–205, 2002.

[11] B. Williams, D. Mehta, T. Camp, and W. Navidi, “Predictive Models to Rebroadcast in Mobile Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, 3(3): 295-303, 2004.

[12] H. Lim and C. Kim, “Flooding in Wireless Ad Hoc Networks,” *Computer Communication Journal*, 24(3-4): 353-363, 2001.

[13] V. Chvatal, “A Greedy Heuristic for the Set-Covering Problem,” *Mathematics of Operation Research*, 4(3): 233-235, 1979.

[14] G. Calinescu, I. Mandoiu, P. Wan, and A. Zelikovsky, “Selecting Forwarding Neighbors in Wireless Ad Hoc Networks,” *ACM Mobile Networks and Applications*, 9: 101-111, 2004.

[15] W. Peng and X. C. Lu, “On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks,” In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 129-130, 2000.

[16] I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks,” *IEEE Trans. on Parallel and Distributed Systems*, 13(1): 14-25, 2002.

[17] W. Peng and X. C. Lu, “AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks,” *Journal of Computer Science and Technology*, 16(2): 114-125, 2001.

[18] A. Qayyum, L. Viennot, and A. Laouiti, “Multipoint Relaying for Flooding Broadcast Message in Mobile Wireless Networks,” In *Proceedings of Hawaii International Conference on System Sciences*, 2002.

[19] W. Lou and J. Wu, “Localized Broadcasting in Mobile Ad Hoc Networks Using Neighbor Designation,” In *Handbook of Mobile Computing*, CRC Press, 2005.

[20] I. Stojmenovic and J. Wu, “Broadcasting and Activity-Scheduling in Ad Hoc Networks,” In *Ad Hoc Networking* , S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., IEEE Press, 205-229, 2004.

[21] X.Y. Li and I. Stojmenovic, "Broadcasting and Topology Control in Wireless Ad Hoc Networks," In *Handbook of Algorithms for Mobile and Wireless Networking and Computing*, A. Boukerche and I. Chlamtac, Eds., CRC Press, 2005.

[22] M. Cagalj, J. Hubaux, and C. Enz, "Minimum-Energy Broadcast in All-wireless Networks: NP-completeness and Distribution Issues," In *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[23] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," In *Proceedings of INFOCOM*, 585-594, 2002.

[24] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized Minimum-Energy Broadcasting in Ad-Hoc Networks," In *Proceedings of INFOCOM*, 2003.

[25] G. Ding, Z. Sahinoglu, B. Bhargava, P. Orlik, and J. Zhang, "Reliable Broadcast in ZigBee Networks," In *Proceedings of SECON*, 2005.

Appendix A

A.1 Proof of Theorem 1

Proof: Assume that there is a node v which is never covered. This implies that all its 1-hop neighbors in $N(v)$ do not rebroadcast after receiving the broadcast packet, or have not received the packet either. We first prove that all the tree neighbors $TN(v)$ of v are not covered either. As long as the total number of nodes is larger than one, $TN(v)$ can not be empty because 1) When v is a root, it must have at least one child; 2) When v is not the root, it must have a parent in order to be connected to the network. For any node $u \in TN(v)$, if it receives the packet but does not rebroadcast, it must have learnt that $v \in TN(u)$ has already been covered. This contradicts the assumption that v is not covered. Hence u must have not received the packet yet.

For every node $u \in TN(v)$, we apply the above argument and find that $TN(u)$ is not covered yet. Hence $TN_2(v)$ is not covered. Similarly, when we apply the above argument to every node in $TN_2(v)$, we further prove that $TN_3(v)$ is not covered. Since any node is no more than $2d_m$ hops away from node v, we can repeat this process up to $2d_m$ times until we find that all nodes in the ZigBee network are not covered, which is obviously not true at least to the broadcast source. So it is proven that every node v in the ZigBee network must be covered as far as the network is connected.

A.2 Proof of Theorem 2

Proof: We first prove that the forward node set $F(v)$ selected by the step 2 and 3 in OOS algorithm will on-tree cover C, and $F(v)$ has the minimum size among all possible forward node sets. Due to the fact that $TN(v) \subseteq N(v)$ and $N(X) - N(Y) \subseteq N(X - Y)$ for any two sets X and Y, it can be easily proven that the set S defined in (1) and (3) can on-tree cover the set C defined in (2) and (4), respectively, and $S \cap C = \emptyset$. Given S and C defined in step 1 of the OOS algorithm, step 2 constructs a group of trees, called a *forest*, of $S \cup C$, which is part of the whole logical ZigBee tree. Since these trees are separated, every node in a tree cannot on-tree cover or be on-tree covered by any node in another tree. So we only need to consider each tree separately.

In step 3, for the bottom level d_m of a tree, from left to right, every node $w \in C$ is processed. Due to $C \subseteq TN(S)$ and node w does not have any children, w's parent x must belong to S and should be selected to on-tree cover w. After processing the bottom level, only nodes in S are left at this level and a minimum number of necessary forward nodes are selected.

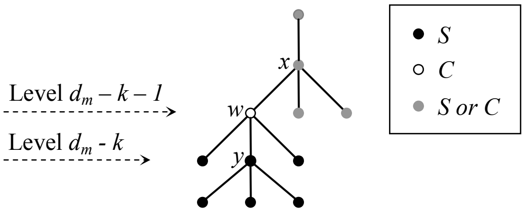


Figure A.1. Step 3 in OOS algorithm

To apply mathematical induction, we assume level d_m up to level $d_m - k$ ($k \geq 0$) have been processed and a minimum number of forward nodes have been selected. Now at level $d_m - k - 1$, for any node $w \in C$, its descendents at lower levels, if any, must belong to S. Since $C \subseteq TN(S)$, w can be on-tree covered by either its parent $x \in S$ or one of its children, say $y \in S$ (Fig. A.1). We would select x if $x \in S$, and we select y only when $x \notin S$. The reason is that, in both cases, one forward node must be selected to on-tree cover w. When $x \in S$ and selected, it will on-tree cover its parent and all its children, including w, if they belong to C. If y is selected, it is only used to cover w because all its children in C have already been processed and removed from C. So we select y only when $x \notin S$. Hence after processing level $d_m - k - 1$, only a minimum number of forward nodes are selected. By mathematical induction, it is finally proven that a minimum size forward node set $F(v)$ is selected from S to on-tree cover C.

For the complexity of the algorithm, since every node in C of size $O(nn_m)$ is visited once, and S of size $O(n)$ is searched in every visit, the total cost will be $O(nn_m\log(n))$, provided S is pre-sorted with complexity of $O(n\log(n))$. Hence the computational complexity of the OOS algorithm is $O(nn_m\log(n))$.

A.3 Proof of Theorem 3

Proof: In the first step of ZOS algorithm, given an arbitrary source node v, it selects its forward nodes set $F(v)$ using sets S and C defined in (1) and (2), respectively. Then node v broadcasts the packet along with $F(v)$ to its 1-hop neighbors in $N(v)$. This transmission certainly covers all 1-hop tree neighbors of v. The non-forward nodes in $N(v) - F(v)$ receive the broadcast packet from v, but drop it. Every node in $F(v)$, however, will run the ZOS algorithm, using sets S and C defined in (3) and (4), respectively, to find its own forward nodes and rebroadcast the packet. As a result, any node in $TN_2(v)$ (actually $TN(N(v))$) will be covered by either $F(v)$ or v itself.

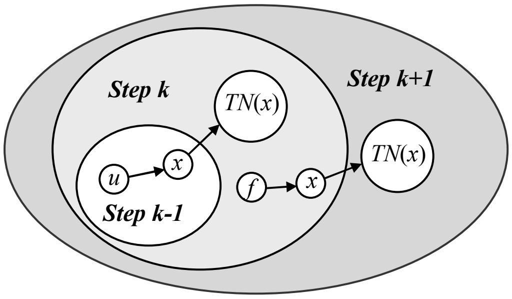


Figure A.2. Step $k + 1$ in the proof of Theorem 3

The same process is repeated by the selected forward nodes (which are not previously non-forward nodes). To apply mathematical induction, we assume that at the end of step $k \geq 2$, every node in $TN_k(v)$ is covered. We next prove that $TN_{k+1}(v)$ will be covered after step $k + 1$. In step $k + 1$, we only need to consider any node $x \in TN_k(v) - TN_{k-1}(v)$. Since x has already been covered, there are two possible ways that x is covered in step k and earlier: 1) If x is covered by a forward node u in step $k - 1$ or before, that is $x \in N(u)$, then $TN(x) - x$ has already been covered in step k or before by either the forward nodes selected by u or u itself; 2) If x is covered in step k by a forward node f selected in step $k - 1$, f has selected its own forward nodes $F(f)$ in step k that will rebroadcast in order to on-tree cover $TN(x) - x$ in step $k + 1$. These two cases are both illustrated in Fig. A.2. So at the end of step $k + 1$, $TN(TN_k(v) - TN_{k-1}(v))$ will be covered. Knowing that

$$TN(TN_k(v) - TN_{k-1}(v)) \supseteq TN_{k+1}(v) - TN_k(v),$$

we prove that all nodes in $TN_{k+1}(v)$ will be covered at the end of step $k + 1$. Finally we have proven that after each step, one more hop of tree neighbors will be covered. Since there are at most $2d_m$ hops between any two nodes on a tree of maximal depth d_m, the total number of steps will be at most $2d_m$. Since the network is connected, every node will eventually be covered and marked as either a forward node or a non-forward node.

Biographies

Gang Ding received his B.E. in control engineering from Tsinghua University, China. He is now a Ph.D. student in ECE Purdue University. His research interests include mobile wireless communications and networking, distributed systems, and control theory. During his PhD study, Gang spent about one year working at Motorola Labs, Mitsubishi Electric Research Labs (MERL), and Intel Research. He has two US patents pending.

Zafer Sahinoglu received his B.S. in EE from Gazi University, Turkey, M.S. in biomedical engineering and PhD (with awards) in EE from New Jersey Institute of Technology (NJIT) in 1998 and 2001 respectively. He worked at AT&T Shannon Research Labs in 1999, and has been with MERL since 2001. His current research interests include NWK and MAC layer issues in wireless sensor networks and impulse radio ultra wide band precision ranging and positioning. He has co-authored a book-chapter on UWB geo-location, and has been author and co-author of more than 30 conference and journal articles; and has provided significant contributions to emerging MPEG-21 standards on mobility modeling and characterization for multimedia service adaptation, to the ZigBee standard on data broadcasting, routing and application profile development, to IEEE 802.15.4a standard on precision ranging. He is currently the chair of ZigBee Industrial Plant Monitoring Application Profile Task Group and the vice technical editor of IEEE 802.15.4a ranging part. He holds one European and six US patents, and has 20 pending.

Philip Orlik received his B.E., M.S., and Ph.D. in electrical engineering from the State University of New York at Stony Brook. He joined MERL in 2000, and is currently a principle technical staff. His research interests include wireless and optical communications, networking, queuing theory, and analytical modeling.

Jinyun Zhang received her Ph.D. in electrical engineering from the University of Ottawa, where she was also a visiting scholar and worked on digital image processing. Prior to this she was a teacher/lecturer at Tsinghua University, China. Jinyun worked for Nortel Networks for the past ten years where she held engineering and most recently management positions in the areas of VLSI design, advanced wireless technology development, wireless networks and optical networks. She has a broad technical background, specializing in system design, DSP algorithms, and real-time embedded S/W for wireless communications and DWDM optical networks. Jinyun joined the MERL in 2001, and is now the manager of Digital Communications and Networking Lab.

Bharat Bhargava received his B.E. from the Indiana Institute of Science and the M.S. and Ph.D. in EE from Purdue University. He is now a professor of computer sciences at Purdue University. His research involves mobile wireless networks, host authentication and key management, secure routing and dealing with malicious hosts, adaptability to attacks, and experimental studies. He has been awarded the charter Gold Core Member distinction by the IEEE Computer Society for his distinguished service. In 1999, he received an IEEE Technical Achievement award for a major impact of his decade long contributions to foundations of adaptability in communication and distributed systems.