

## Research Article

# Extension of Pairwise Broadcast Clock Synchronization for Multicluster Sensor Networks

Kyoung-Lae Noh,<sup>1</sup> Yik-Chung Wu,<sup>2</sup> Khalid Qaraqe,<sup>3</sup> and Bruce W. Suter<sup>4</sup>

<sup>1</sup> Digital Solution Center, Corporate Technology Operations, Samsung Electronics Co., Ltd., South Korea

<sup>2</sup> Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong

<sup>3</sup> Texas A&M University at Qatar, P.O. Box 23874, Doha, Qatar

<sup>4</sup> Information Directorate, Air Force Research Laboratory/RITC, Rome, NY 13441, USA

Correspondence should be addressed to Yik-Chung Wu, [ycwu@ieee.org](mailto:ycwu@ieee.org)

Received 26 April 2007; Revised 28 September 2007; Accepted 15 November 2007

Recommended by Paul Cota

Time synchronization is crucial for wireless sensor networks (WSNs) in performing a number of fundamental operations such as data coordination, power management, security, and localization. The Pairwise Broadcast Synchronization (PBS) protocol was recently proposed to minimize the number of timing messages required for global network synchronization, which enables the design of highly energy-efficient WSNs. However, PBS requires all nodes in the network to lie within the communication ranges of two leader nodes, a condition which might not be available in some applications. This paper proposes an extension of PBS to the more general class of sensor networks. Based on the hierarchical structure of the network, an energy-efficient pair selection algorithm is proposed to select the best pairwise synchronization sequence to reduce the overall energy consumption. It is shown that in a multicluster networking environment, PBS requires a far less number of timing messages than other well-known synchronization protocols and incurs no loss in synchronization accuracy. Moreover, the proposed scheme presents significant energy savings for densely deployed WSNs.

Copyright © 2008 Kyoung-Lae Noh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. INTRODUCTION

Recently, a huge attention has been paid to wireless sensor networks (WSNs) as fundamental infrastructures for future ubiquitous communication environments [1, 2]. With the help of current technical developments in microelectromechanical systems (MEMS) and wireless communications, the feasibility of WSNs keeps rapidly growing. Time (clock) synchronization is a procedure for providing a common notion of time across a distributed system. Hence, it is essential to maintain data consistency and coordination, and to perform other fundamental operations [2–4]. The Network Time Protocol (NTP) [5] is the most popular synchronization protocol for distributed networks due to its diverse advantages in the Internet environment. However, NTP is subject to a number of critical issues when applied to WSNs because of the unique nature of sensor networks: limited power resources, adverse wireless channel conditions, and dynamic topology changes. For this reason, different types of synchronization schemes have

been developed thus far for sensor network applications [2].

The Reference-Broadcast Synchronization (RBS) protocol was proposed to synchronize a group of wireless sensors within the transmission range of the reference sensor node, which alleviates the effects of random delays in timing message delivery [6]. Using a similar approach to NTP, the Timing-sync Protocol for Sensor Networks (TPSN) was proposed in [7]. TPSN is based on the level hierarchy of the network, and synchronizes the entire network by exchanging timing messages along every branch (edge) of the hierarchical tree. For synchronization protocols based on the two-way message exchanges like TPSN, a family of energy-efficient clock offset and skew (frequency offset) estimators was recently proposed in [8].

More recently, the Flooding Time Synchronization Protocol (FTSP) [9] synchronizes the network by successively broadcasting the synchronization messages using MAC layer time-stamping and performing skew compensation based on linear regression. The Time Diffusion Protocol (TDP) was

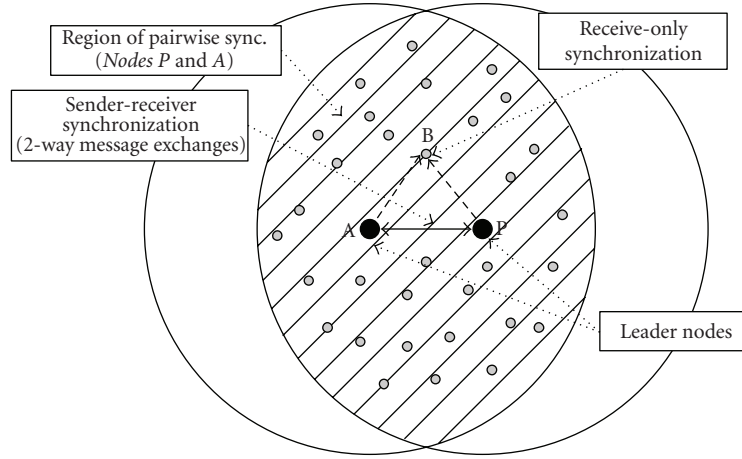


FIGURE 1: Pairwise broadcast synchronization for a single-cluster network.

proposed in [10]. TDP selects a set of the diffusion leaders in every level of the network considering the balance of workload and the stability of the local clocks. Considering uniformly distributed quantization noise, *Sadler* derived the joint maximum likelihood clock offset and skew estimators, and also proposed a detection mechanism of clock drift [11]. *Giridhar* and *Kumar* proposed a distributed clock synchronization algorithm to improve the accuracy level of synchronization under the condition that every connected edge exchanges timing messages [12]. Besides, several time synchronization protocols based on the beacon transmission at the physical layer have been reported as well. Assuming a realistic wireless channel environment, a distributed broadcasting time synchronization scheme was proposed by *Khajehnouri* and *Sayed* to overcome the effects of multipath frequency selective fading in [13]. A low-complexity bio-inspired synchronization protocol for large scale WSNs was reported by *Hong* and *Scaglione* in [14].

The tradeoff between the accuracy and energy consumption (complexity) is the most important and crucial factor in designing time synchronization protocols for WSNs due to the space and power limitations of sensor nodes. Indeed, more energy consumption is required in general to increase the synchronization accuracy. Hence, the energy consumption for synchronization should be kept as small as possible while satisfying a certain accuracy level. The Pairwise Broadcast Synchronization (PBS) protocol was recently proposed with the aim of minimizing the overall energy consumption for achieving global network synchronization without incurring any loss in synchronization accuracy relative to the existing protocols [15]. PBS is based on the idea that while two nodes performing synchronization using two-way message exchanges, other nodes lying nearby can overhear the messages and can also synchronize themselves. PBS efficiently combines the merits of two different basic synchronization approaches, namely, the sender-receiver synchronization (SRS) and the receiver-only synchronization (ROS) approaches, to achieve global synchronization with a significantly reduced number of synchronization messages, that is, with reduced energy consumption. However, the original

form of PBS assumes that every node in the network should be located within the communication ranges of the leader nodes. That is, PBS is mainly designed for single-cluster sensor networks, and hence, its efficient extension to general multicluster-based sensor networks represents an interesting open research problem. This paper studies a multicluster extension of PBS based on the level hierarchy of the network and proposes an energy efficient pair selection algorithm to achieve global synchronization.

The rest of this paper is organized as follows. In Section 2, we overview the key features of PBS and illustrate the way to achieve networkwide synchronization for single-cluster sensor networks. For the extension to general multicluster sensor networks, Section 3 proposes the networkwide pair selection algorithm and the groupwise pair selection algorithm to select the best synchronization sequence aiming at minimizing the overall energy consumption. Besides, Section 3 presents simulation results on the performance of the proposed pair selection algorithms with respect to the number of required synchronization messages (i.e., energy consumption). Finally, Section 4 summarizes and concludes this paper.

## 2. SYNCHRONIZATION FOR SINGLE-CLUSTER NETWORKS USING PAIRWISE BROADCAST SYNCHRONIZATION

Suppose there are two leader nodes (*Nodes P* and *A*) in the network, and every node in the network is located within the communication ranges of these leader nodes as depicted in Figure 1. Note that the leader nodes are just ordinary nodes like other sensor nodes in the network. Here, the network consists of a single cluster, and the two leader nodes perform a pairwise synchronization using two-way timing message exchanges, which has been thoroughly analyzed in [8, 17, 18]. Note that all the nodes in the common coverage (checked) region can receive a series of synchronization messages containing information about the time stamps of the pairwise synchronization. Using this information, any node in the checked region can also be synchronized to *Node P* by

applying the ROS approach with no additional timing message transmissions [15]. Here, *Nodes P* and *A* provide synchronization beacons for all the nodes located in their vicinity.

More specifically, the clock model for PBS is described in Figure 2, where  $\theta_{\text{offset}}^{(AP)}$  stands for the clock offset between *Nodes A* and *P*, and  $\theta_{\text{offset}}^{(BP)}$  is the clock offset between *Nodes B* and *P*. In order to synchronize *Nodes A* and *P*, *Node A* transmits a synchronization packet to *Node P*, which contains the level and identifier (ID) of *Node A* and the values of time stamp  $T_{1,i}^{(A)}$ . *Node P* receives it at  $T_{2,i}^{(P)}$  and transmits an acknowledgment packet to *Node A* at  $T_{3,i}^{(P)}$ . This packet contains the level and ID of *Node P* and the values of time stamps  $T_{1,i}^{(A)}$ ,  $T_{2,i}^{(P)}$ , and  $T_{3,i}^{(P)}$ . Finally, *Node A* receives the acknowledgment packet at  $T_{4,i}^{(A)}$ . The above timing messages exchange procedure is performed multiple ( $N$ ) times, and the clock offset and skew between *Nodes P* and *A* can be estimated based on  $T_{1,i}^{(A)}$ ,  $T_{2,i}^{(P)}$ ,  $T_{3,i}^{(P)}$ , and  $T_{4,i}^{(A)}$  [8].

Now, consider an arbitrary node, say *Node B*, in Figure 1. While *Nodes P* and *A* are exchanging time messages, *Node B* is capable of receiving packets from both nodes. At *Node B*, when it receives packets from *Node A*, it records the arrival time as  $\{T_{2,i}^{(B)}\}_{i=1}^N$ , as shown in Figure 2. Similarly, when *Node B* receives packets from *Node P*, the arrival time is recorded as  $\{T_{2,i}^{(P)}\}_{i=1}^N$ . Besides, *Node B* can also get the time readings  $\{T_{1,i}^{(A)}\}_{i=1}^N$  since it is embedded in the packets from *Node A*. Based on the time readings  $\{T_{1,i}^{(A)}\}_{i=1}^N$ ,  $\{T_{2,i}^{(B)}\}_{i=1}^N$ , and  $\{T_{2,i}^{(P)}\}_{i=1}^N$  in *Node B*, the joint clock offset and skew estimator using the ROS approach is given by [15]

$$\begin{aligned} \begin{bmatrix} \hat{\theta}_{\text{offset}}^{(BP)} \\ \hat{\theta}_{\text{skew}}^{(BP)} \end{bmatrix} &= \frac{1}{N \sum_{i=1}^N D_i^2 - \left[ \sum_{i=1}^N D_i \right]^2} \\ &\times \begin{bmatrix} \sum_{i=1}^N D_i^2 \sum_{i=1}^N x[i] - \sum_{i=1}^N D_i \sum_{i=1}^N [D_i \cdot x[i]] \\ N \sum_{i=1}^N [D_i \cdot x[i]] - \sum_{i=1}^N D_i \sum_{i=1}^N x[i] \end{bmatrix}, \end{aligned} \quad (1)$$

where  $D_i \triangleq T_{1,i}^{(A)} - T_{1,1}^{(A)}$  and  $x[i] \triangleq T_{2,i}^{(P)} - T_{2,i}^{(B)}$ . Consequently, *Node B* can be synchronized to *Node P* using the results in (1), and all the other nodes in the common coverage region in Figure 1 can also be simultaneously synchronized to *Node P* without any additional timing message transmissions, thus saving a significant amount of energy. Note that it was shown in [15] that the synchronization accuracy of PBS is exactly the same as the RBS protocol.

For a network with  $L$  sensor nodes, let  $N_{\text{TPSN}}$ ,  $N_{\text{FTSP}}$ , and  $N_{\text{RBS}}$  denote the numbers of required timing messages in network synchronization using TPSN, FTSP, and RBS, respectively. It has been proven [16] that  $N_{\text{TPSN}} = 2N(L-1)$ ,  $N_{\text{FTSP}} = NL$ , and  $N_{\text{RBS}} = N + L(L-1)/2$ , where  $N$  is the number of times synchronization messages are transmitted or exchanged when synchronizing two nodes.

It is remarkable that the required number of timing messages for all the above-mentioned protocols is proportional

to the number of sensors in the network  $L$  or its square  $L^2$ . On the other hand, since PBS adopts the energy efficient ROS approach, it can synchronize a set of nodes based on the messages exchanged between the two leader nodes. Thus PBS requires only  $2N$  timing messages during each synchronization period (i.e.,  $N_{\text{PBS}} = 2N$ ). Hence,  $N_{\text{PBS}}$  does not depend on the number of sensors in the network, a fact which incurs an enormous amount of energy saving. Moreover, this gain increases proportionally with respect to the scale of the network. Consequently, the benefit of PBS over RBS, TPSN, and FTSP is clear and huge in terms of energy consumption.

### 3. SYNCHRONIZATION FOR MULTICLUSTER NETWORKS

In the previous section, we only concentrated on the case where all the nodes lie within a single cluster. For example, in Figure 1, all the nodes are located inside the checked region. In this section, we will present the extension of PBS to networks which consist of more than one cluster.

In a multicluster network, there are two possible scenarios for extending the proposed PBS. When there is no problem with the deployment of leader nodes in the right positions of the network, the whole sensor field can be divided into several clusters, where each cluster contains two individual leader nodes whose communication ranges cover the entire cluster. Hence, every cluster can be first synchronized by performing a pairwise synchronization between the pair of leader nodes and other nodes within the cluster performing ROS. Then, like RBS, global synchronization can be achieved by additional message exchanges (based on SRS) among leader nodes in different clusters. In this case, the extension of PBS becomes mostly the problem of network implementation just like cell-planing problems in mobile communication networks.

However, if deploying leader nodes in a planned fashion is not possible, then there is no way to apply the above-mentioned procedure. For this general scenario, we have to choose which nodes perform pairwise synchronization and which nodes perform ROS. For the rest of the paper, we focus on this scenario since it represents a more general situation. Considering the energy-efficiency requirement in time synchronization, the question becomes how to select the optimum set of nodes that performs pairwise synchronizations such that all other nodes in the network can be synchronized using ROS?

In this paper, we propose an energy-efficient pair selection algorithm, named the groupwise pair selection algorithm (GPA), to achieve global synchronization using ROS. In the following, we first show a way to achieve global synchronization based on the networkwide heuristic search in order to reveal some preliminary ideas on pair selection problem. Then, the proposed GPA is presented in detail.

#### 3.1. Networkwide pair selection algorithm

Considering the energy efficiency in time synchronization, the problem of finding the optimum set of pairwise synchronizations is equivalent to that of minimizing the number of

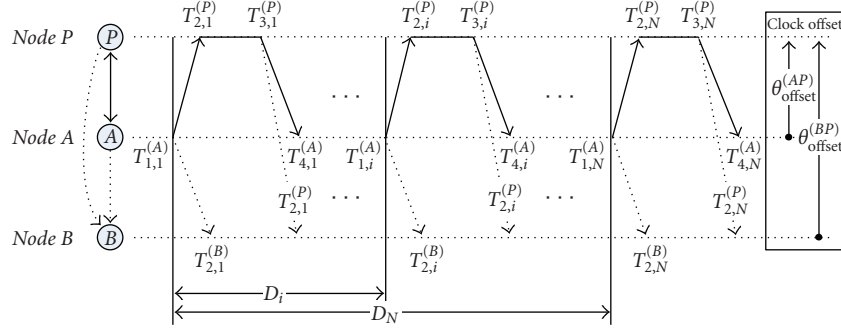


FIGURE 2: Clock synchronization model of PBS.

overall pairwise synchronizations in the network. There are two fundamental criteria to select the best synchronization pairs as follows:

- (1) a pair of nodes containing the maximum number of nodes in their common coverage region of the pairwise synchronization has to be chosen during each selection step of the synchronization pair;
- (2) a pair of nodes in the same level should not be selected as a valid pair in order to limit the bound for the maximum synchronization error which increases with the number of levels of synchronization.

Therefore, to find the best synchronization pairs, information about the network hierarchy and connectivity, which can be obtained by beacon exchanges among nodes, is required. This can be accomplished by applying the well-known breath-first search algorithm [21], in which every node in the network is required to send messages with their maximum power satisfying a certain energy constraint.

For a graphical illustration of the proposed algorithms, Figure 3 shows an example of a network connection hierarchy. The pairwise synchronization begins with the reference node *Node 1*, and four different branches (edges) are connected to the reference, that is, there are four different nodes which can be chosen as the first synchronization pair. As mentioned before, the criterion for selecting the best pair is to find a pair of nodes maximizing the number of synchronizing nodes (based on the ROS approach) from the pairwise synchronization. Let  $p_{i,j}$  denote the pairwise synchronization between *Nodes i* and *j*, and let  $\mathbf{p}$  represent the pairwise synchronization sequence vector whose elements are a set of  $p_{i,j}$ . Define also, by  $N_{ROS}^{i,j}$ , the number of synchronizing nodes, which are performing ROS from  $p_{i,j}$ . In Figure 3, *Node 4* must be selected as the first pair node since  $N_{ROS}^{1,4} = 3$ , and it represents the maximum achievable value among all possible choices (all the other nodes in level 1, *Nodes 2, 3, and 5*, can be synchronized from  $p_{1,4}$ ). The same criterion can be applied to determine the next pair of nodes thereafter, until all the nodes in the network are synchronized. Therefore,  $p_{3,8}$ ,  $p_{4,11}$ , and  $p_{11,14}$  are chosen as the second, third, and fourth pairs, respectively. Consequently, a sequence of pairwise synchronizations is chosen to maximize the number of nodes performing ROS. In this example, the pairwise synchronization sequence vector is given by  $\mathbf{p} = \{p_{1,4}, p_{3,8}, p_{4,11}, p_{11,14}\}$ .

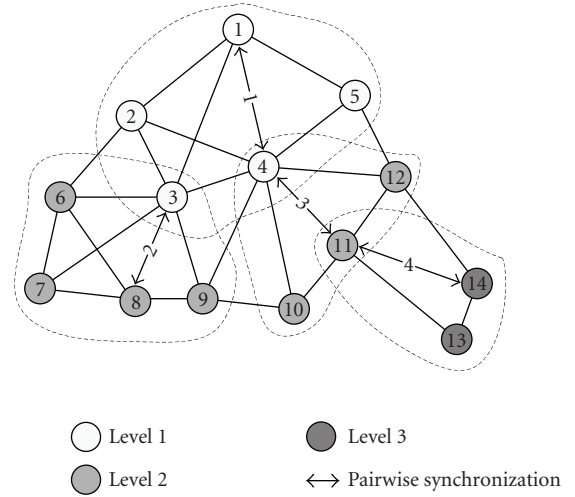


FIGURE 3: Network connection hierarchy for networkwide pair selection algorithm.

Now, we formally present the Networkwide Pair Selection Algorithm (NPA) to find the pairwise synchronization sequence. A network can be represented as a graph  $G = (V, E)$ , where  $V$  represents the set of nodes (e.g., in Figure 3,  $V = \{s_i\}_{i=1}^{14}$ ) and  $E$  stands for the set of edges (branches), whose elements are 2-element subsets of  $V$ . Assume  $L_i$  denotes the subset of nodes located on level  $i$  (e.g.,  $L_0 = \{s_1\}$ ,  $L_1 = \{s_i\}_{i=2}^{14}$ ,  $L_2 = \{s_i\}_{i=6}^{12}$ , and  $L_3 = \{s_{13}, s_{14}\}$  for the example depicted by Figure 3). Let  $S$  denote the set of synchronized nodes whose initial element is  $S = \{s_1\}$ , and let  $M_{i,j}$  denote the  $i$ th row and  $j$ th column element of the adjacency matrix  $M$  of the graph  $G$ , where  $M_{i,j} = 1$  when *Nodes i* and *j* are connected, and  $M_{i,j} = 0$  otherwise.

Note that an arbitrary node *Node k* can be synchronized from  $p_{i,j}$  if and only if *Nodes i* and *j* are connected and *Node k* is connected to both *Nodes i* and *j*, that is,  $M_{i,j} = M_{i,k} = M_{j,k} = 1$ . Besides, the levels of the nodes in a synchronization pair must differ by one. Therefore, the number of synchronizing nodes from  $p_{1,i}$  ( $N_{ROS}^{1,i}$ ) is given by

$$N_{ROS}^{1,i} = \sum_{j \neq i} M_{1,i} \cdot M_{1,j} \cdot M_{i,j} \quad \forall s_i, s_j \notin S, \quad \forall s_i, s_j \in L_1 \quad (2)$$



Hence, the first node to perform pairwise synchronization with  $s_1$  can be obtained by maximizing  $N_{ROS}^{1,i}$  as follows:

$$\hat{i} = \arg \max_i N_{ROS}^{1,i}, \quad (3)$$

where  $s_i \in L_1$ , otherwise, no connection exists between *Nodes* 1 and  $i$ . In the example of Figure 3,  $\hat{i} = 4$  because  $N_{ROS}^{1,4} = 3$  and achieves the maximum value (note that if there are multiple candidates that maximize  $N_{ROS}^{1,i}$ , the algorithm chooses randomly among these candidates). Thus,  $p_{1,4}$  is selected as the first pair. Note that because of the second selection criterion mentioned above, in general, to find the second pair of nodes in this example, another node in  $L_1$  should be chosen until all the nodes in  $L_1$  are synchronized. However, in this example, there are no remaining unsynchronized nodes in  $L_1$  after  $p_{1,4}$  since all the nodes in  $L_1$  are already synchronized by  $p_{1,4}$  (i.e.,  $S = \{L_0, L_1\}$  after the first pairwise synchronization).

The same maximization procedure can be applied to find the next synchronization pair. A general formula for finding  $N_{ROS}^{i,j}$  is given by

$$N_{ROS}^{i,j} = \sum_{k \neq j} M_{i,j} \cdot M_{i,k} \cdot M_{j,k} \quad \forall s_i \in S, s_j, s_k \notin S, \quad (4)$$

where  $s_i$  is a candidate of the next parent node and the levels of  $s_j$  and  $s_k$  are different from those of the parent node by one in accordance with the second selection criterion. The next synchronization pair can be found by maximizing  $N_{ROS}^{i,j}$  as follows:

$$(\hat{i}, \hat{j}) = \arg \max_{i,j} N_{ROS}^{i,j}. \quad (5)$$

Here,  $p_{\hat{i},\hat{j}}$  becomes the next element of  $\mathbf{p}$  and all synchronized nodes from  $p_{\hat{i},\hat{j}}$  are added to  $S$ . From (4) and (5), the second synchronization pair becomes  $p_{3,8}$  in this example since  $N_{ROS}^{3,8} = 4$  and is maximum among all possible combinations of  $i$  and  $j$ . Thus,  $\mathbf{p}$  becomes  $\{p_{1,4}, p_{3,8}\}$  and  $S = \{L_0, L_1, \{s_i\}_{i=6}^9\}$ . Likewise, the third pair is chosen to be  $p_{4,11}$ ,  $\mathbf{p} = \{p_{1,4}, p_{3,8}, p_{4,11}\}$ , and  $S = \{L_0, L_1, L_2\}$ . Repeating the same procedure (with  $s_i \in L_2$ ) yields  $p_{11,14}$  as the last synchronization pair, and hence, a complete sequence becomes  $\mathbf{p} = \{p_{1,4}, p_{3,8}, p_{4,11}, p_{11,14}\}$  as depicted in Figure 3. Figure 4 summarizes the NPA.

### 3.2. Groupwise pair selection algorithm

To discover the overall network connectivity, every single node in the network has to transmit the connection discovery beacons and send back acknowledgment packets upon receiving other beacons from its adjacent nodes (e.g., the breath-first search algorithm in [21]). For WSNs consisting of a large number of nodes, discovering the network connectivity is not a simple task and requires a huge number of packet exchanges. Therefore, we propose an efficient alternative method, the Groupwise Pair Selection Algorithm (GPA), which relies on the hierarchical structure (spanning tree) of the network to simplify the connection discovery procedure.

**Input:** Graph ( $G$ ), Adjacency matrix ( $M$ ),  
Maximum level/depth ( $d_{\max}$ )  
**Output:** PS sequence vector ( $\mathbf{p}$ )  
**Initial values:**  $n = m = 1$ ,  $S = \{s_1\}$   
1 **while**  $n \leq d_{\max} - 1$  **do**  
2   **while**  $\exists s_j \in L_n$  and  $s_j \notin S$  **do**  
3     **for all**  $i, j$ , and  $k$  with  
       $s_i \in S, s_i \in L_{n-1}, \{s_j, s_k\} \notin S$ , and  $\{s_j, s_k\} \in L_n$   
4        $N_{ROS}^{i,j} \leftarrow \sum_{k \neq j} M_{i,j} \cdot M_{i,k} \cdot M_{j,k}$   
5        $(\hat{i}, \hat{j}) \leftarrow \arg \max_{i,j} N_{ROS}^{i,j}$   
6        $\mathbf{p}(m) \leftarrow p_{\hat{i},\hat{j}}$   
7        $m \leftarrow m + 1$   
8       All synchronized nodes from  $p_{\hat{i},\hat{j}}$  are added to  $S$   
9     **end while**  
10     $n \leftarrow n + 1$   
11 **end while**  
\*  $\mathbf{p}(m)$ :  $m$ th element of  $\mathbf{p}$

FIGURE 4: Networkwide pair selection algorithm.

Note that the hierarchical tree of the network can be generated by a level discovery procedure as discussed in [7]. Once a hierarchical tree is established, there exist groups of nodes, where a group consists of a parent and its children nodes, for example, in Figure 5(a), *Nodes* 1, 2, 3, 4, and 5 form a group with *Node* 1 being the parent and other nodes being children. Similarly, another example is *Nodes* 3, 6, 7, 8, and 9 form another group with *Node* 3 being the parent node and other nodes being the children nodes. Two additional groups in this example are *Nodes* 4, 10, 11, 12 and *Nodes* 11, 13, 14, respectively.

In GPA, instead of discovering the entire network connectivity, every parent node only investigates the connectivity among its children nodes (detailed procedure is to be presented in the next section). Therefore, the reference node does not need to find the pairwise synchronization sequence of the entire network, but only needs to find the pairwise synchronization sequence among its children, and the other parent nodes successively perform the same connection searching procedure as the reference node. As a result, GPA significantly reduces the complexity of building up a connection hierarchy, and requires a far smaller number of connection discovery beacons than NPA due to its limited set of scanning nodes.

Once the hierarchy of the whole network and the connectivity within every group of nodes have been established, the children nodes in each group synchronize with the parent node using either pairwise synchronization or ROS. In other words, the problem of synchronizing the whole network reduces to synchronizing a number of individual groups, where each group consists of a parent and a number of children. In order to minimize the total number of synchronization messages for the whole network, it is equivalent to minimizing the number of timing message exchanges in each group.

For each group  $i$ , assume the parent node is *Node*  $i$ . Further, let  $\mathbf{p}_i$  represent the pairwise synchronization sequence

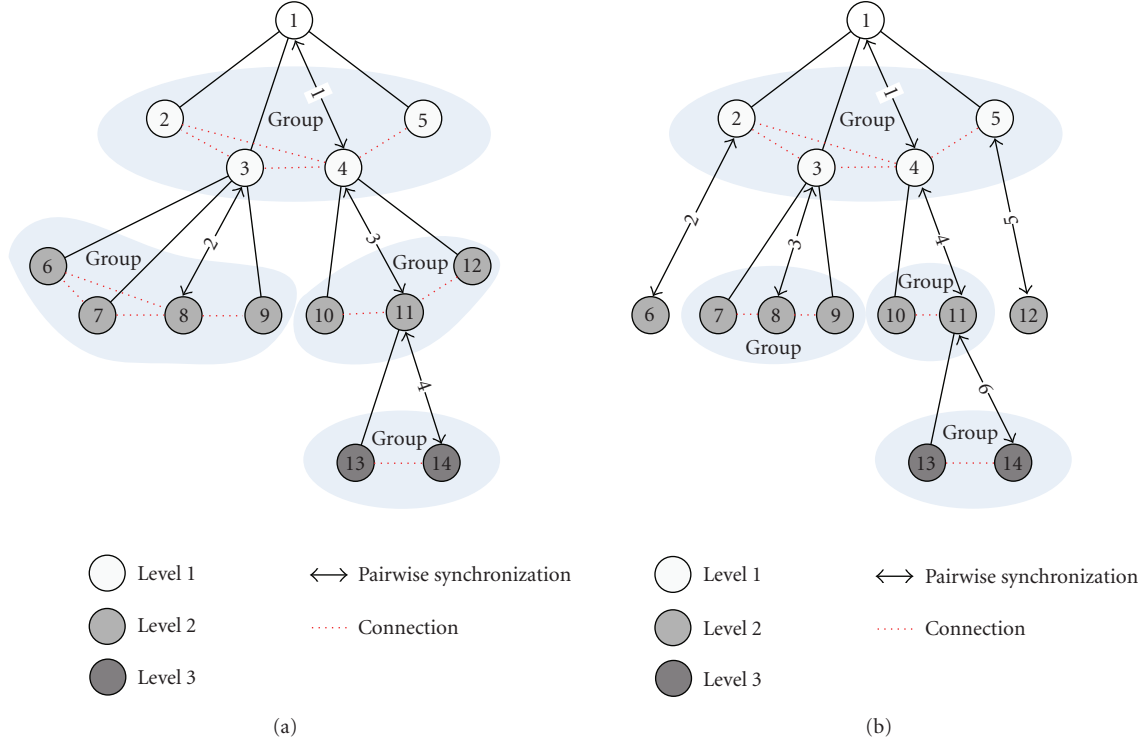


FIGURE 5: Examples of hierarchical spanning trees for groupwise pair selection algorithm.

for group  $i$  and let  $S_i$  denote the set of synchronized nodes in group  $i$  with the initial element  $S_i = \{s_i\}$ . The number of synchronizing nodes from  $p_{i,j}$  is given by

$$N_{ROS}^{i,j} = \sum_{k \neq j} M_{j,k} \quad \forall s_j, s_k \notin S_i. \quad (6)$$

In order to minimize the number of message exchanges in group  $i$ , the first child node chosen for pairwise synchronization with its parent should be

$$\hat{j} = \arg \max_j N_{ROS}^{i,j}. \quad (7)$$

In this way, the maximum number of children nodes can be synchronized using ROS. After that, all synchronized nodes from  $p_{i,\hat{j}}$  are added to  $S_i$ , and  $p_{i,\hat{j}}$  is added to  $\mathbf{p}_i$ . If there is any node in group  $i$  left unsynchronized, (6) and (7) are repeated until all nodes are synchronized. In the example of Figure 5(a), Nodes 4, 8, 11, and 14 are chosen to perform pairwise synchronization with their respective parents. The proposed GPA for group  $i$  is summarized in Figure 6.

It is obvious that in GPA, the workload for finding the best pairwise synchronization sequence is shared among the reference node and the other parent nodes, that is, no networkwide heuristic connectivity search is required for GPA. Notice that in the example of Figure 5(a), the network synchronized using GPA requires the same number of pairwise synchronizations as that of NPA. However, the number of pairwise synchronizations for GPA depends on the specific

hierarchical tree, which is randomly constructed, and in general, is greater than that of NPA. For instance, for another possible tree of the network as in Figure 5(b), the required number of pairwise synchronizations is 6 instead of 4. Although it is true that, in general, GPA requires additional synchronization messages relative to NPA; in the next section, we will show by simulations that this difference is very small. On the other hand, the savings in complexity for establishing the network hierarchy in GPA significantly outweighs the slight increase in terms of the number of synchronization messages, when compared to NPA. Next, we will present the connection discovery process for GPA.

### 3.2.1. Groupwise connection discovery

As the level discovery phase in TPSN [7], GPA first creates a hierarchical structure (spanning tree) of the network, then it searches the connection status among a set of children nodes in every parent-children group. The connection discovery procedure in GPA consists of the following steps:

- (1) select a reference node using an appropriate leader election algorithm (or picks up a node having the highest priority) and assign it to level zero;
- (2) the reference node broadcasts a level discovery packet containing the identity and the level of packet;
- (3) every node who receives a level discovery packet assigns its level in increasing order and sends a new level discovery packet attaching its own level; after being assigned a level, every node discards further packets requesting level discovery to prevent collision;

**Input:** The connectivity information  $M_{j,k}$  for all  $s_j, s_k$  within group  $i$   
**Output:** PS sequence vector ( $\mathbf{p}_i$ ) of group  $i$   
**Initial value:**  $m = 1, S_i = \{s_i\}$  where  $s_i$  is the parent node of group  $i$

```

1  while  $\exists s_j \in \text{group } i \text{ and } s_j \notin S_i$  do
2    for all  $j$  and  $k$  s.t.  $\{s_j, s_k\} \in \text{group } i$ , and  $\{s_j, s_k\} \notin S_i$ 
3       $N_{\text{ROS}}^{i,j} \leftarrow \sum_{k \neq j} M_{j,k}$ 
4       $\hat{j} \leftarrow \arg \max_j N_{\text{ROS}}^{i,j}$ 
5       $\mathbf{p}_i(m) \leftarrow p_{i,\hat{j}}$ 
6       $m \leftarrow m + 1$ 
7      All synchronized nodes from  $p_{i,\hat{j}}$  are added to  $S_i$ 
8  end while
*  $\mathbf{p}_i(m)$ :  $m$ th element of  $\mathbf{p}_i$ 

```

FIGURE 6: Groupwise pair selection algorithm for each group  $i$ .

- (4) repeat (5) until every node in the network is successfully assigned a level;
- (5) once a hierarchical tree is established, every parent-children group performs the following operations: every child node broadcasts a connection discovery packet to other children nodes and sends back acknowledgment packets upon receiving other connection discovery packets; connection discovery packets from any child node belonging to other groups will be discarded.

Notice that other algorithms (e.g., [22, 23]) can also be considered when constructing the spanning tree (i.e., steps (1)–(4) above).

Figure 7 compares the complexity of NPA in establishing the network connection hierarchy with that of GPA, which assumes a level hierarchy, with respect to the number of sensor nodes. In this simulation, sensors are randomly deployed in the area  $100 \times 100$ , the transmission range of each sensor is set to be 25, and the reference node is assumed to be located at the center of the simulation area. 100,000 network topologies are generated and the average complexity result is presented. It can be seen that the complexity becomes greater as the number of sensor nodes (equivalently the density) increases. The number of required discovery messages for NPA is about four times larger than that of GPA. The following section analyzes the proposed algorithms in terms of the number of synchronization timing messages, and compares them with the existing protocols.

*Remark 1.* In this paper, we do not consider mobile sensor networks, but fixed sensor networks. Therefore, reconstruction of network hierarchy is not (or rarely) required after the initial connection discovery. Moreover, according to the simulation results in Figures 7 and 8 (to be presented in the next section), the required number of messages for discovering network hierarchy in GPA is comparable to that of only a single synchronization round. Consequently, the overhead of constructing network hierarchy is not significant and negligible for fixed sensor network applications.

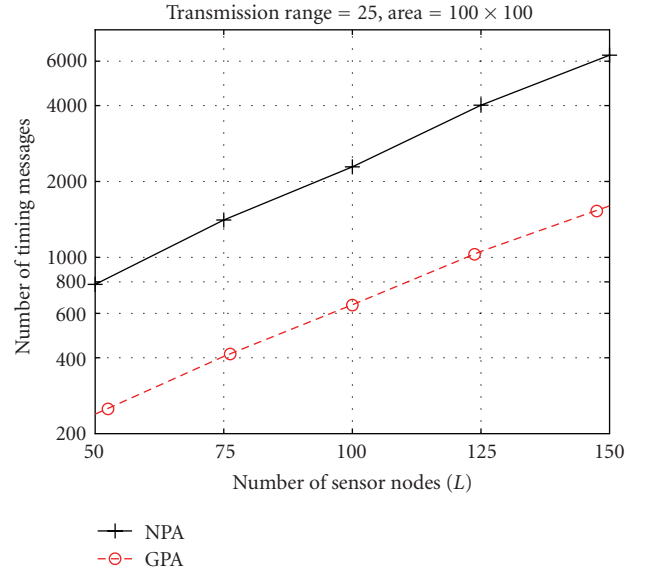


FIGURE 7: Number of messages for constructing the network hierarchy (GPA versus NPA).

### 3.3. Comparisons and analysis

This section compares the proposed algorithms with other conventional protocols such as TPSN, RBS, and FTSP in terms of the number of required synchronization timing messages in order to predict the energy required for network-wide synchronization. Assume that  $|\mathbf{p}|$  denotes the number of elements in a pairwise synchronization sequence vector  $\mathbf{p}$ , then the total number of timing messages for NPA ( $N_{\text{NPA}}$ ) is given by

$$N_{\text{NPA}} = 2N|\mathbf{p}|, \quad (8)$$

where  $N$  is the number of beacons in each pairwise synchronization. Similarly, for GPA, the total number of timing

messages ( $N_{\text{GPA}}$ ) is given by

$$N_{\text{GPA}} = 2N \sum_{i=1}^{N_G} |\mathbf{p}_i|, \quad (9)$$

where  $N_G$  denotes the number of parent-children groups and  $\mathbf{p}_i$  denotes the pairwise synchronization sequence vector of the  $i$ th group. In the given example,  $|\mathbf{p}| = 4$  (see Figure 3) and  $\sum_{i=1}^{N_G} |\mathbf{p}_i| = 4$  or 6 (see Figures 5(a) and 5(b)), that is,  $N_{\text{NPA}} = 8N$  and  $N_{\text{GPA}} = 8N$  or  $12N$ . Notice that in the given example, while  $|\mathbf{p}_i| = 1$  for all  $i$ , there might exist situations that  $|\mathbf{p}_i| > 1$  for some other networks.

In Figure 8, the performances of  $N_{\text{NPA}}$  and  $N_{\text{GPA}}$  are compared with that of  $N_{\text{TPSN}}$  and  $N_{\text{RBS}}$  with respect to the number of overall sensor nodes. Again, in this simulation, the sensor nodes are randomly deployed on an area of  $100 \times 100$ , the transmission range of each sensor is 25, and the reference node is assumed to be located at the center of the simulation area. The number of beacons ( $N$ ) is set to be 10 in this simulation. It can be seen that PBS (with both GPA and NPA) requires a much lower number of timing messages than the other protocols, such as TPSN, FTSP, and RBS, and the gaps between the required number of message transmissions of PBS and those of other protocols become greater as  $L$  increases. Therefore, for densely deployed WSN, PBS has a significant benefit in terms of energy consumption versus either TPSN or RBS. Besides, the proposed GPA performs quite close to NPA, even though it does not require a heuristic network connection search. As mentioned before, GPA can be implemented by simply adding a groupwise connection discovery procedure to the conventional level discovery process in an arbitrary level-based synchronization protocol like TPSN.

Figure 9 evaluates the performance of the proposed algorithms with respect to the transmission range of sensor nodes assuming the same simulation setup as in the previous figure. The number of overall sensor nodes is fixed to 100 in this simulation. It can be seen that as the transmission range (density of the network) increases,  $N_{\text{GPA}}$  decreases since more sensor nodes are able to perform ROS.

*Remark 2.* Although the number of required messages is not a complete measure to represent the overall energy consumption of the network, comparing radio transmission complexity is meaningful enough to evaluate the energy efficiency since, in general, message transmission requires the largest amount of energy consumption among all possible states of a sensor node.

In [24], the authors predict energy consumption of a sensor node based on a Markovian model with respect to the node state, where the idle state requires 0.01 mW, the active listening state requires 1 mW, and the transmission state requires 10 mW, respectively. Hence, message transmission consumes a magnitude greater power than message reception, and a thousand times greater power than keeping the idle state.

As another example, [25] examined the current consumption for transmitting a single radio message at maximum transmit power on the Mica2 mote. It was shown that

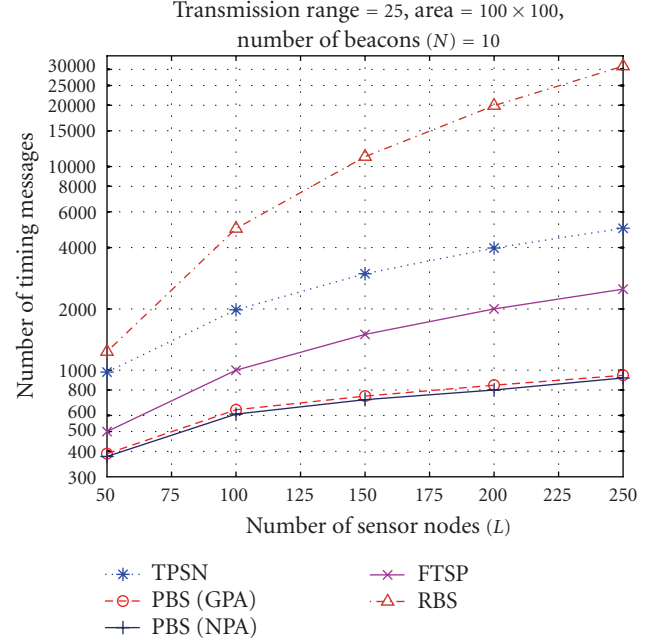


FIGURE 8: Required number of message exchanges with respect to the number of sensor nodes.

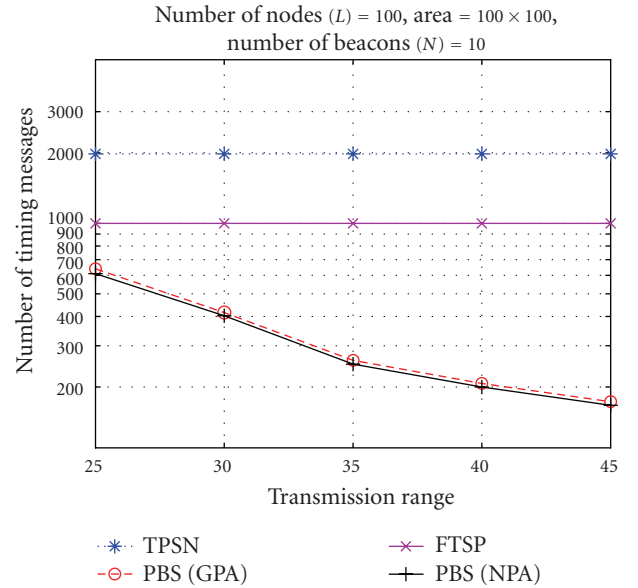


FIGURE 9: Required number of message exchanges with respect to the transmission range.

the idle state consumes instantly  $100 \mu\text{A}$ , the listening state consumes instantly  $10 \text{ mA}$ , and the transmission state consumes instantly  $25 \text{ mA}$ , respectively. In addition, for Mica2 mote, transmitting a message also requires the mote to listen to the radio channel to detect potential collision before beginning transmission. Thus, message transmission simultaneously requires extra power for listening when using the CSMA/CA mechanism.



Note that there exist other models suggesting that energy consumed in idle listening or eavesdropping can be significant compared with the energy required for transmission, depending upon the transmission range and radio environment. In this paper, we have not considered these models. Detailed energy analysis of the proposed schemes is deferred for future investigation.

**Remark 3.** The synchronization accuracy is another crucial designing factor to be concerned with. In general, it depends on a variety of factors, such as the network platform and setup, channel status, and estimation schemes. The performance of existing protocols has been compared in terms of the synchronization accuracy in various references (e.g., [1, 3, 9, 19]). As proven in [15, 16], the accuracy of PBS is exactly the same as that of RBS. Therefore, the issue of synchronization accuracy is not discussed in this paper.

#### 4. CONCLUSIONS

In this paper, a novel time synchronization protocol has been proposed to reduce the overall energy consumption in synchronization based on the receiver-only synchronization approach. In the Pairwise Broadcast Synchronization (PBS) protocol, a number of sensor nodes can be synchronized by only overhearing time message exchanges between pairs of nodes. Thus, PBS significantly reduces the overall network-wide energy consumption by decreasing the number of required timing messages in synchronization.

For networks consisting of multiple clusters, PBS first investigates a hierarchical connection tree of the network, then applies an energy-efficient pair selection algorithm, named groupwise pair selection algorithm (GPA), to achieve global synchronization. The proposed GPA only searches the connectivity among children nodes in every parent-children group of the spanning tree. Moreover, GPA can be easily combined with other level-based protocols by simply adding a groupwise connection discovery procedure. PBS requires a far smaller number of timing messages than other well-known protocols such as RBS, TPSN, and FTSP, and the benefits of this scheme remarkably increase as the number of sensors increases or the sensors are densely deployed.

The proposed new scheme could be fully or partially applied to improve the performance (energy efficiency) of existing protocols or for designing new protocols. Experimental performance evaluation and comparisons with other existing protocols represent an open research work for the future.

#### REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] N. Bulusu and S. Jha, Eds., *Wireless Sensor Networks: A Systems Perspective*, Artech House, Norwood, Mass, USA, 2005.
- [3] B. M. Sadler and A. Swami, "Synchronization in sensor networks: an overview," in *Proceedings of IEEE Military Communications Conference (MILCOM '06)*, pp. 1–6, Washington, DC, USA, October 2007.
- [4] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [5] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
- [6] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pp. 147–163, Boston, Mass, USA, December 2002.
- [7] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 138–149, ACM Press, Los Angeles, Calif, USA, November 2003.
- [8] K.-L. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 766–777, 2007.
- [9] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 39–49, ACM Press, Baltimore, Md, USA, November 2004.
- [10] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 384–397, 2005.
- [11] B. M. Sadler, "Local and broadcast clock synchronization in a sensor node," *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 9–12, 2006.
- [12] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: algorithms and analysis," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 4915–4920, San Diego, Calif, USA, December 2006.
- [13] N. Khajehnouri and A. H. Sayed, "A distributed broadcasting time-synchronization scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '05)*, vol. 5, pp. 1053–1056, Philadelphia, Pa, USA, March 2005.
- [14] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 1085–1099, 2005.
- [15] K.-L. Noh and E. Serpedin, "Pairwise broadcast clock synchronization for wireless sensor networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '07)*, pp. 1–6, Helsinki, Finland, June 2007.
- [16] K.-L. Noh, Y.-C. Wu, K. Qaraqe, and E. Serpedin, "Time synchronization for wireless sensor networks," in *Adaptive Signal Processing for Wireless Communications*, M. Ibnkahla, Ed., CRC Press, 2008.
- [17] V. Paxson, "On calibrating measurements of packet transit times," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '98)*, vol. 26, no. 1, pp. 11–21, Madison, Wis, USA, June 1998.
- [18] D. R. Jeske, "On maximum-likelihood estimation of clock offset," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 53–54, 2005.

- [19] Z. Tian, X. Luo, and G. B. Giannakis, "Cross-layer sensor network synchronization," in *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers (ACSSC '04)*, vol. 1, pp. 1276–1280, Pacific Grove, Calif, USA, November 2004.
- [20] A.-S. Hu and S. D. Servetto, "Asymptotically optimal time synchronization in dense sensor networks," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA '03)*, pp. 1–10, San Diego, Calif, USA, September 2003.
- [21] B. Awerbuch and R. Gallager, "A new distributed algorithm to find breadth first search trees," *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 315–322, 1987.
- [22] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless, "Minimum power configuration in wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 390–401, Urbana-Champaign, Ill, USA, May 2005.
- [23] J. van Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA '03)*, pp. 11–19, San Diego, Calif, USA, September 2003.
- [24] M. Achir and L. Ouvry, "Power consumption prediction in wireless sensor networks," in *Proceedings of the 16th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Antwerp, Belgium, August-September 2004.
- [25] V. Shnayder, M. Hempstead, B.-R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, Baltimore, Md, USA, November 2004.