Efficient Training of Sensor Networks

A.A. Bertossi[1], S. Olariu[2], and M.C. Pinotti[3]

[1] Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7,
40127 Bologna, Italy, bertossi@cs.unibo.it
[2] Department of Computer Science, Old Dominion University, Norfolk,
VA 23529-0162, USA, olariu@cs.odu.edu
[3] Department of Computer Science and Mathematics, University of Perugia,
06123 Perugia, Italy, pinotti@unipg.it

Abstract. Due to their small form factor and modest energy budget, individual sensors are not expected to be GPS-enabled. Moreover, in most applications, exact geographic location is not necessary, and all that the individual sensors need is a coarse-grain location awareness. The task of acquiring such a coarse-grain location awareness is referred to as training. In this paper, a scalable energy-efficient training protocol is proposed for massively-deployed sensor networks, where sensors are initially anonymous and unaware of their location. The training protocol is lightweight and simple to implement; it is based on an intuitive coordinate system imposed onto the deployment area which partitions the anonymous sensors into clusters where data can be gathered from the environment and synthesized under local control.

1 Introduction

Recent advances in nano-technology have made it feasible to develop miniaturized low-power devices that integrate sensing, special-purpose computing and wireless communications capabilities [1, 12, 19]. These small devices, commonly called *sensors*, will be mass-produced, making their production cost negligible. A sensor has a small, non-renewable power supply and, once deployed, must work unattended. A massive deployment of sensors, perhaps in the order of thousands or even tens of thousands [17], is expected.

Aggregating sensors into sophisticated computational and communication infrastructures, called *wireless sensor networks*, will have a significant impact on a wide array of applications ranging from military, to scientific, to industrial, to health-care, to domestic, establishing ubiquitous wireless sensor networks that will pervade society redefining the way in which we live and work [12]. The novelty of wireless sensor networks and the tremendous potential for a multitude of application domains has triggered a lot of activity in both academia and industry [3–5, 7, 9].

The peculiar characteristics of sensor networks (a massive deployment of sensors, the anonymity of individual sensors, a limited battery power budget per sensor, and a possibly hostile environment) pose unique challenges to the

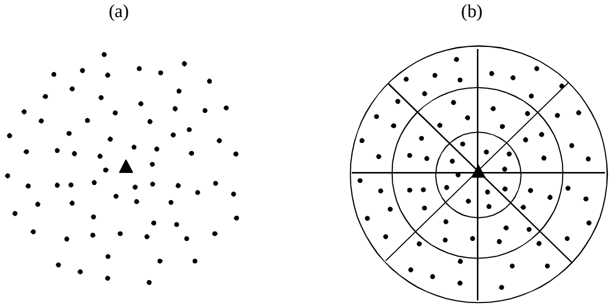

Fig. 1. (a) *A sensor network with a central sink node.* (b) *The trained sensor network.*

design of protocols. First of all, the limited power budget requires the design of ultra-lightweight communication protocols. However, how data collected by sensors are queried and accessed and how concurrent sensing can be performed internally are of significance as well. An important guideline in this direction is to perform as much local data processing at the sensor level as possible, avoiding the transmission of raw data through the sensor network. This implies that the sensor network must be multi-hop and only a small number of sensors have the sink as one of their one-hop neighbors. For reasons of scalability, it is assumed that no sensor knows the topology of the network.

Several possible techniques can be used for interfacing sensor networks to the outside world and for harvesting the data they produce. The simplest technique involves using one or several sink nodes, i.e. special long-range radios deployed alongside with the sensors. Each sink has a full range of computational capabilities, can send long-range directional broadcasts to the sensors at distance at most R, can receive messages from nearby sensors, and has a steady power supply. In this scenario, the raw data collected by individual sensors are fused, in stages, and forwarded to the nearest sink that provides the interface to the outside world. Such a scenario for a sensor network with a single central sink is depicted in Figure 1(a).

There are some applications requiring sensory data with exact geographical location, motivating the development of communication protocols that are location aware and perhaps location dependent. In some other applications, however, exact geographic location is not necessary, and all that the individual sensors need is only coarse-grain location awareness [11, 16]. Of course, there is a trade-off, because coarse-grain location awareness is lightweight but the resulting accuracy is only a rough approximation of the exact geographic coordinates.

The random deployment results in sensors initially unaware of their location. Further, due to limitations in form factor, cost per unit and energy budget, individual sensors are not expected to be GPS-enabled. Moreover, many probable application environments limit satellite access. Therefore, individual sensors have to determine their exact geographic location, if required by the application, or else a coarse-grain approximation thereof. The former task is referred to as *localization* and has been extensively studied in the literature [8, 10]. The latter task, referred to as *training*, has been considered by Olariu et al. [11, 16, 18]. In particular, they devised two elegant training protocols for sensor networks, which differ on whether or not sensors need some kind of synchronization with the sink. Such two training protocols have different performance, measured in terms of overall time for training, sensor awake time, and number of sensor wake/sleep transitions.

The main contribution of the present paper is to present a new training protocol which outperforms that originally presented in [16], lowering its overall time for training from a linear to a square-root function of the size of the coordinate system used for location awareness. The protocol assumes that the sink and the sensors are somehow synchronized [11, 13, 14].

The remainder of this paper is organized as follows. Section 2 discusses the sensor model used throughout the work. Section 3 introduces the task of training, that is, endowing individual sensors with coarse-grain location awareness. Training imposes a coordinate system which divides the sensor network area into equiangular wedges and concentric coronas centered at the sink, as first suggested in [16]. Section 4 is the backbone of the entire paper, presenting the theoretical underpinnings of the training protocol. In the corona training protocol, time is ruled into slots and each sensor has to learn a string of bits representing its corona number. The protocol consists of two phases. The first phase is centralized and sink-driven. Its computation can be thought as a visit of a complete binary tree, whose leaves represent coronas, whose vertex preorder numbers are related to the time slots, and whose vertex inorder numbers are related to the transmission range used by the sink. At the end of the first phase, sensors that belong to a group of some consecutive coronas have learned the same most significant bits. The second phase is distributed and, within each group, the sensors that have already known their corona number inform those in the next corona to properly set their remaining bits. Finally, Section 5 offers concluding remarks.

2 The sensor model

We assume a sensor to be a device that possesses three basic capabilities: sensory, computation, and wireless communication. The sensory capability is necessary to acquire data from the environment; the computational capability is necessary for aggregating data, processing control information, and managing both sensory and communication activity. Finally, the wireless communication capability is necessary for sending/receiving aggregated data and control information to/from other sensors or the sink.

We assume that individual sensors are tiny, mass-produced devices that operate subject to the following fundamental constraints.

a. Sensors are *anonymous* – they do not have individually unique IDs;
b. Each sensor has a modest non-renewable energy budget;
c. In order to save energy, sensors are in *sleep* mode most of the time, waking up for short intervals;
d. Each sensor has a modest transmission range, perhaps a few meters – this implies that out-bound messages can reach only the sensors in its proximity, typically a small fraction of the sensors deployed;
e. Individual sensors must work *unattended* – once deployed it is either infeasible or impractical to devote attention to individual sensors.

It is worth mentioning that while the energy budget can supply short-term applications, sensors dedicated to work over extended periods of time may need to scavenge energy from the specific environment they are placed into, employing light, temperature, vibration, kinetics, magnetic fields, etc.

3 Training a sensor network

In this work we assume a wireless sensor network that consists of a sink and a set of sensors randomly deployed in its broadcast range as illustrated in Figure 1(a). For simplicity, we assume that the sink is centrally placed, although this is not really necessary.

The task of endowing sensors with coarse-grain location awareness, referred to as training, is essential in several applications. One example is *clustering* where the set of sensors deployed in an area is partitioned into clusters [1, 2, 6, 15]. As a result of training, we impose a coordinate system onto the sensor network in such a way that each sensor belongs to exactly one cluster. The coordinate system divides the sensor network area into equiangular wedges. In turn, these wedges are divided into sectors by means of concentric circles or coronas centered at the sink and whose radii are determined to optimize the transmission efficiency of sensors-to-sink transmission. Sensors in a given sector map to a cluster, the mapping between clusters and sectors being one-to-one. In particular, a cluster is the locus of all nodes having the same coordinates in the coordinate systems [11].

Referring to Figure 1(b), the task of training a sensor network involves establishing [16]:

1. *Coronas*: The deployment area is covered by k coronas determined by k concentric radii $0 < r_1 < r_2 < \cdots < r_k = R$ centered at the sink.
2. *Wedges*: The deployment area is ruled into a number of angular wedges, centered at the sink, which are established by directional transmission [11].

As illustrated in Figure 1(b), at the end of the training period each sensor has acquired two coordinates: the identity of the corona in which it lies, as well as the identity of the wedge to which it belongs. Importantly, the locus of all the sensors that have the same coordinates determines a cluster.

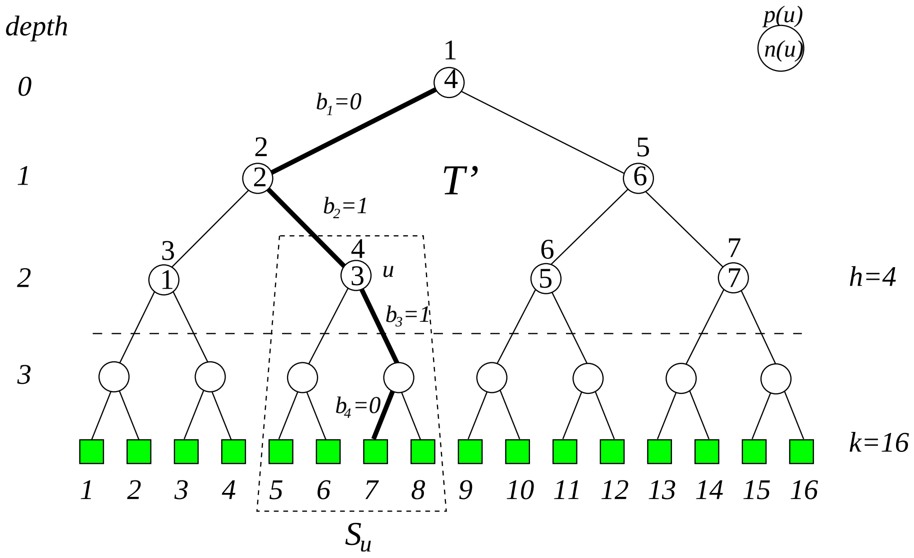

Fig. 2. *Illustrating corona training. Labels outside nodes of T' give their preorder numbers, while labels inside give their inorder numbers. Leaves represent coronas, numbered from* 1 *to* k.

4 The corona training protocol

The main goal of this subsection is to present the details of the corona training protocol. The wedge training protocol is similar (in fact, simpler) and will not be discussed.

Let k be an integer known to the sensors and let the k coronas be determined by concentric circles of radii $r_1 < r_2 < \cdots < r_k$ centered at the sink node. For simplicity we shall assume that k is a power of two. Given the transmission range r of each sensor, the radius r_i is assumed to be equal to ir, namely the corona width is $r_{i+1} - r_i = r$.

The idea of the corona training protocol is for each individual sensor to learn the identity of the corona to which it belongs. For this purpose, each individual sensor learns a string of $\log k$ bits from which the corona number can be determined easily. To see how this is done, it is useful to assume time ruled into slots and that the sensors can synchronize to the master clock running at the sink node.

In time slot s_1 all the sensors are awake and the sink transmits with a power level corresponding to $r_{\frac{k}{2}}$. In other words, in the first slot the sensors in the first $\frac{k}{2}$ coronas will receive the message above a certain threshold, while the others will not. Accordingly, the sensors that receive the signal set $b_1 = 0$, the others set $b_1 = 1$.

Consider a k-leaf complete binary tree T, whose leaves are numbered left to right from 1 to k. The edges of T are labeled by 0's and 1's in such a way that an edge leading to a left-subtree is labeled by a 0 and an edge leading to a right subtree is labeled by a 1. Let ℓ, $(1 \leq \ell \leq k)$, be an arbitrary leaf and let $b_1, b_2, \ldots, b_{\log k}$ be the edge labels of the unique path leading from the root to ℓ. It is both well known and easy to prove by a standard inductive argument that

$$\ell = 1 + \sum_{j=1}^{\log k} b_j 2^{\log k - j}. \tag{1}$$

For example, refer to Figure 2, where $k = 16$. Applying Equation (1) to leaf 7, we have: $7 = 1 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$.

Let h be an integer known to the sensors which is a power of two such that $1 \leq h \leq k/2$. Consider the subtree T' consisting of the uppermost $2h - 1$ nodes of T. Refer again to Figure 2, where $h = 4$ and T' consists of the uppermost 7 nodes. Let u be an arbitrary node in T', other than the root, and let $b_1, b_2, \ldots, b_i$ be the edge labels on the unique path from the root to u, where i is the depth of u in T' and $1 \leq i \leq \log h$. Note that the root of T' is at depth $i = 0$, and it is characterized by an empty sequence of edge labels.

We take note of the following technical result.

Lemma 1. *Let u be an arbitrary node of depth i in T'. Then, the preorder number $p(u)$ of u is given by*

$$p(u) = 1 + \sum_{j=1}^{i} c_j$$

where

$$c_j = \begin{cases} 1 & \text{if } b_j = 0 \\ \frac{h}{2^{j-1}} & \text{if } b_j = 1. \end{cases}$$

Proof. The proof was first given in [16], and it is reported here for the sake of completeness. The proof is by induction on the depth i of node u in T'. To settle the basis, note that for $i = 0$, u must be the root and $p(u) = 1$, as expected.

For the inductive step, assume the statement true for all nodes in T' of depth less than the depth of u. Indeed, let v be the parent of u and consider the unique path of length i joining the root to u. Clearly, nodes u and v share $b_1, b_2, \ldots, b_{i-1}$ and, thus, $c_1, c_2, \ldots, c_{i-1}$. By the inductive hypothesis,

$$p(v) = 1 + \sum_{j=1}^{i-1} c_j. \tag{2}$$

On the other hand, since v is the parent of u, we can write

$$p(u) = p(v) + \begin{cases} 1 & \text{if } u \text{ is the left child of } v \\ \frac{h}{2^{i-1}} & \text{otherwise} \end{cases} \tag{3}$$

Notice that if u is the left child of v we have $b_i = 0$ and $c_i = 1$; otherwise $b_i = 1$ and $c_i = \frac{h}{2^{i-1}}$. This observation, along with (2) and (3) combined, allows us to write

$$p(u) = 1 + \sum_{j=1}^{i-1} c_j + c_i = 1 + \sum_{j=1}^{i} c_j$$

completing the proof of the lemma.

As an example, consider node u in Figure 2. Applying Lemma 1, one gets $p(u) = 1 + 1 + \frac{4}{2^1} = 4$.

Lemma 2. *Let u be an arbitrary node of depth i in T'. Then, the inorder number $n(u)$ of u is given by*

$$n(u) = h + \sum_{j=1}^{i} d_j$$

where

$$d_j = \begin{cases} -\frac{h}{2^j} \text{ if } b_j = 0 \\ +\frac{h}{2^j} \text{ if } b_j = 1 \end{cases}$$

Proof. The proof is by induction on the depth i of node u in T'. To settle the basis, note that for $i = 0$, u must be the root and $n(u) = h$, as expected.

For the inductive step, assume the statement true for all nodes in T' of depth less than the depth of u. Indeed, let v be the parent of u and consider the unique path of length i joining the root to u. Clearly, nodes u and v share $b_1, b_2, \dots, b_{i-1}$ and, thus, $d_1, d_2, \dots, d_{i-1}$. By the inductive hypothesis,

$$n(v) = h + \sum_{j=1}^{i-1} d_j. \tag{4}$$

On the other hand, since v is the parent of u, we can write

$$n(u) = n(v) + \begin{cases} -\frac{h}{2^i} \text{ if } u \text{ is the left child of } v \\ +\frac{h}{2^i} \text{ otherwise} \end{cases} \tag{5}$$

Notice that if u is the left child of v we have $b_i = 0$ and $d_i = -\frac{h}{2^i}$; otherwise $b_i = 1$ and $d_i = \frac{h}{2^i}$. This observation, along with (4) and (5) combined, allows us to write

$$n(u) = h + \sum_{j=1}^{i-1} d_j + d_i = h + \sum_{j=1}^{i} d_j$$

completing the proof of the lemma.

As an example, consider again node u in Figure 2. Applying Lemma 2, one gets $n(u) = 4 - \frac{4}{2^1} + \frac{4}{2^2} = 3$.

With these technicalities out of the way, we now return to the corona training protocol. In our setting, the leaves of T represent the k coronas, while the preorder and inorder numbers of the nodes in T' are related, respectively, to the time slots in the training protocol and to the transmission ranges used by the sink. The goal of the training protocol is that all the sensors belonging to any corona c have to learn the $\log k$ bits, $b_1, b_2, \ldots, b_{\log k}$, which are the binary representation of their corona number minus one.

The corona training protocol consists of two phases: a first centralized phase, followed by a second distributed phase. The first phase is sink-driven and lasts for $2h-1$ time slots. During this phase, the sensors learn the leftmost $\log h+1$ bits, while the remaining bits will be learned in the second phase. At each time slot of the first phase, the sink transmits with a suitable power level and some sensors are awake to learn one more bit. The procedures performed by the sink node and the awake sensors in the centralized phase are described as follows.

Referring again to the T' tree, consider a generic time slot s_z, with $1 \leq z \leq 2h-1$. Let u be the vertex of T' such that its preorder number $p(u)=z$, and let S_u be the subtree of T rooted at u (see e.g. Figure 2). At time slot s_z, the sink node transmits with a power level equal to $r_{\frac{k}{2h}n(u)}$, where $n(u)$ is the inorder number of node u, and the awake sensors are those belonging to the coronas which are the leaves of S_u. Although all the sensors in the coronas $1, \ldots, \frac{k}{2h}n(u)$ can hear the sink transmission, only those awake will learn one more bit. Precisely, the awake sensors that hear the sink transmission get $b_{i+1}=0$, while the awake sensors that do not hear anything get $b_{i+1}=1$, where i is the depth of node u in T'. It is worthy to note that, at time slot s_1, u is the root of T' and thus all the sensors are awake. As soon as a sensor has learned b_{i+1}, with $i \leq \log h$, it can easily compute the value c_{i+1} given in Lemma 1, and hence derive $p(u)+c_{i+1}$. If $p(u)+c_{i+1}=z+1$ (i.e., $b_{i+1}=0$), the sensor remains awake; otherwise, it goes to sleep. If $i<\log h$, then the sensor will wake up again at time slot $s_{p(u)+c_{i+1}}$. If $i=\log h$, let γ be the integer represented by the $\log h+1$ bits learned so far by the sensor, namely $\gamma=\sum_{j=1}^{\log h+1} b_j 2^{\log h+1-j}$, then the sensor will wake up again at time slot $s_{2h+2\gamma}$.

In order to verify the correctness of the first phase of the corona training protocol, the following lemma is useful.

Lemma 3. *Let u be any node of T', with depth $i>0$, and let v be the parent of u. The subtree S_u rooted at u contains $|n(v)-n(u)|\frac{k}{h}$ leaves, whose indices are:*

$$\begin{cases} (2n(u)-n(v))\,\frac{k}{2h}+1, \;\ldots,\; n(v)\frac{k}{2h} & \text{if } u \text{ is the left child of } v \\ n(v)\frac{k}{2h}+1, \;\ldots,\; (2n(u)-n(v))\,\frac{k}{2h} & \text{if } u \text{ is the right child of } v \end{cases}$$

Proof. It follows immediately by the definition of the inorder number $n(u)$ and by the fact that any subtree S_u, with root at depth $i=\log h$, has $\frac{k}{h}$ leaves of T.

As an example, refer again to Figure 2, where the labels outside the nodes of T' give their preorder numbers, while those inside give their inorder numbers.

Consider the node u having $p(u) = 4$ and $n(u) = 3$. The subtree S_u contains 4 leaves, indexed $5, 6, 7, 8$. Indeed, u is a right child, its parent v has $n(v) = 2$, $|n(v) - n(u)| \frac{k}{h} = |2 - 3| \frac{16}{4} = 4$, $n(v) \frac{k}{2h} + 1 = 2 \frac{16}{8} + 1 = 5$, and $(2n(u) - n(v)) \frac{k}{2h} = (6 - 2) \frac{16}{8} = 8$.

Theorem 1. *Consider a time slot s_z, with $1 \leq z \leq 2h - 1$. At time s_z, all the sensors belonging to any corona c, with $1 \leq c \leq k$, have learned bits $b_1, b_2, \ldots, b_{i+1}$, where i is the depth of the deepest node u on the unique path from the root to leaf c such that $p(u) \leq z$.*

Proof. The proof is by induction on z. As the basis, observe that for $z = 1$, the root of T' is the only node with $p(u) \leq 1$. Observe that the depth of the root u is 0, all the sensors are awake, and the sink has transmitted with a power level $r_{n(u)\frac{k}{2h}} = r_{h\frac{k}{2h}} = r_{\frac{k}{2}}$. Therefore, all the sensors in any corona c have learned bit b_1, namely those in the first $\frac{k}{2}$ coronas have learned 0, and the others have learned 1.

For the inductive step, assume the statement true for $z - 1$. At time slot s_z, the only sensors awake are those belonging to the coronas which are the leaves of the subtree S_u, rooted at the node u such that $p(u) = z$. All the sensors in the other coronas are sleeping. Indeed, this is correct since the deepest node on the unique path from the root has not changed, and therefore such sensors have to learn no bits during this time slot. To check that the sensors in S_u learn the right bit, consider the node v such that $p(v) = z - 1$, and let w be the lowest common ancestor between u and v. Let ℓ be the depth of w. By inductive hypothesis, since $p(w) < p(u)$, all the sensors in S_u already know bits $b_1, \ldots, b_{\ell+1}$. At time s_z, the sink transmits with power level $r_{n(u)\frac{k}{2h}}$. Two cases may arise.

Case 1. Node u is the left child of w, that is $w = v$. By Lemma 3, the index of the middle corona among the leaves of S_u is $(2n(u) - n(v)) \frac{k}{2h} + (n(v) - n(u)) \frac{k}{2h} = n(u) \frac{k}{2h}$. Therefore, the sensors in the coronas $(2n(u) - n(v)) \frac{k}{2h} + 1, \ldots, n(u) \frac{k}{2h}$ learn $b_{\ell+2} = 0$, while those in $n(u) \frac{k}{2h} + 1, \ldots, n(v) \frac{k}{2h}$ learn $b_{\ell+2} = 1$. Since the depth of u is $\ell + 1$, the statement is proved.

Case 2. Node u is the right child of w, and hence $w \neq v$. By Lemma 3, the index of the middle corona among the leaves of S_u is $n(w) \frac{k}{2h} + (n(u) - n(w)) \frac{k}{2h} = n(u) \frac{k}{2h}$. Therefore, the sensors in the coronas $n(w) \frac{k}{2h} + 1, \ldots, n(u) \frac{k}{2h}$ learn $b_{\ell+2} = 0$, while those in $n(u) \frac{k}{2h} + 1, \ldots, (2n(u) - n(w)) \frac{k}{2h}$ learn $b_{\ell+2} = 1$. Since the depth of u is $\ell + 1$, the statement is proved.

To illustrate Theorem 1, refer again to node u of Figure 2. Only the sensors in the leaves of S_u are awake in time slot $s_{p(u)} = s_4$, while the sink node transmits with a range of $r_{n(u)\frac{k}{2h}} = r_6$ since $\frac{k}{2h} = \frac{16}{8} = 2$ and $n(u) = 3$. The sensors in the leaves of S_u at a distance from the sink not exceeding r_6 will receive the signal, while the others will not. Since the depth of u is 2, the sensors in leaves 5 and 6 learn bit $b_3 = 0$, while those in leaves 7 and 8 learn bit $b_3 = 1$.

Corollary 1. *At time slot s_{2h-1}, the first phase of the corona training protocol is completed, and the sensors belonging to any corona c, with $1 \leq c \leq k$, have learned the leftmost $\log h + 1$ bits, $b_1, \ldots, b_{\log h+1}$, of the binary representation of their corona number minus one.*

Consider now the second phase of the corona training protocol, which starts at time slot s_{2h}. During such a phase, all the sensors have to learn the remaining $\log k - \log h - 1$ bits, $b_{\log h+2}, \ldots, b_{\log k}$. Observe that there are $2h$ groups, each of $\frac{k}{2h}$ consecutive coronas which have learned the same $\log h + 1$ bits. Within each group, the sensors that belong to the first and last corona can become aware of their position by listening to the sink. Subsequently, in a distributed way, the sensors that have already known their position can inform those in the next corona to properly set their remaining bits.

The algorithm for the second phase is detailed as follows. Consider a generic group γ consisting of coronas $\gamma\frac{k}{2h} + 1, \ldots, (\gamma + 1)\frac{k}{2h}$, with $0 \leq \gamma \leq 2h - 1$. At the beginning of the second phase, all the sensors in such a group know $\gamma = \sum_{j=1}^{\log h+1} b_j 2^{\log h+1-j}$, and wake up at time slot $s_{2h+2\gamma}$. At time slot $s_{2h+2\gamma}$, the sink transmits with a power level of $r_{(\gamma+1)\frac{k}{2h}-1}$. The sensors that do not hear it set every bit $b_{\log h+2}, \ldots, b_{\log k}$ to 1 and go to sleep. At time slot $s_{2h+2\gamma+1}$, the sink transmits with a power level of $r_{\gamma\frac{k}{2h}+1}$. The sensors that hear it set $b_{\log h+2}, \ldots, b_{\log k}$ to 0, start the distributed computation by sending a message within their local transmission range, and then go to sleep. In a subsequent time slot s_t, an awake sensor that receives a message from another sensor computes $\delta = t-(2h+2\gamma+1)$, sets its bits $b_{\log h+2}, \ldots, b_{\log k}$ to the binary representation of δ (with the most significant bit assigned to $b_{\log_{h+2}}$), and goes to sleep. Therefore, the following result easily holds.

Lemma 4. *All the sensors belonging to corona c, with $1 \leq c \leq k$, have learned the binary representation of $c - 1$ at time slot*

$$\begin{cases} s_{2h+2\gamma+1+\delta} & \textit{if } c = \gamma\frac{k}{2h} + \delta + 1 \\ s_{2h+2\gamma} & \textit{if } c = (\gamma + 1)\frac{k}{2h} \end{cases}$$

where $0 \leq \gamma \leq 2h - 1$ and $0 \leq \delta \leq \frac{k}{2h} - 1$.

The sensors in corona $k - 1$ are the last to learn all their bits. Since they belong to the $\left(\frac{k}{2h} - 1\right)$-th corona of group $\gamma = 2h - 1$, this happens at time $s_{(2h+2(2h-1)+1)+\frac{k}{2h}-1}$. Thus, the entire corona training protocol finishes at time $s_{6h-2+\frac{k}{2h}}$. Therefore, the overall time to accomplish the above corona training protocol is $O(h + \frac{k}{h})$. Such a time is minimized when $h = O(\sqrt{k})$, and in such a case it becomes $O(\sqrt{k})$, improving over the $O(k)$ time of the training protocol presented in [16].

It is also worth noting that only the sensor nodes that need to be awake in a given time slot will stay awake, the others will sleep minimizing the power expenditure. Yet another interesting feature of the training protocol is that individual sensors sleep for as many contiguous time slots as possible before waking up,

Performance Measure	**Complexity**	$h = O(\sqrt{k})$	$h = O\left(\frac{k}{\log k}\right)$
Overall Time	$O\left(h + \frac{k}{h}\right)$	$O(\sqrt{k})$	$O\left(\frac{k}{\log k}\right)$
Sensor Awake Time	$O\left(\log h + \frac{k}{h}\right)$	$O(\sqrt{k})$	$O(\log k)$
♯ Wake/Sleep Transitions	$O(\log h)$	$O(\log k)$	$O\left(\log\left(\frac{k}{\log k}\right)\right)$

Table 1. *Performance of the corona training protocol. The sensor awake time is optimal when* $h = O\left(\frac{k}{\log k}\right)$*, while the overall time is minimized when* $h = O(\sqrt{k})$*.*

thus avoiding repeated wake-sleep transitions that are known to waste energy. To see this, observe that the sensors remain awake $O(\log h)$ time during the first phase because they wake up just at the time slot when they have to learn one more bit. Moreover, the sensors are awake for $O(\frac{k}{h})$ consecutive time slots during the second phase. Therefore, the sensor awake time is $O(\log h + \frac{k}{h})$, which is minimized when $h = O(\frac{k}{\log k})$. In such a case the sensor awake time is $O(\log k)$ which is optimal since every sensor has to learn $\log k$ bits and it cannot learn more than one bit at a time. In addition, the number of wake-sleep transitions is $O(\log h)$. Precisely, referring again to Figure 2, one notes that a wake-sleep transition occurs every time a node u on the path of T' from the root to a generic corona is a right child of its parent. This is because the preorder numbers of u and its parent are not consecutive. Thus, the worst case arises for the sensors in the coronas of group $2h - 1$, which go through exactly $\log h + 1$ transitions during the first phase, plus one more transition during the second phase. The performance achieved by the corona training protocol for the above mentioned measures are summarized in Table 1.

5 Concluding remarks

In this work a new training protocol has been proposed which outperforms that originally presented in [16] in terms of the overall time for training, lowering it from a linear to a square-root function of the size of the coordinate system used for location awareness. However, a number of questions still remain open. In particular, a good idea for further work should be that of comparing the performance of the protocol proposed in the present paper with that devised in [18]. Indeed, while the training protocol of Section 4 requires individual sensors to be awake for a short time but consumes energy in both the synchronization between the sensors and the sink and the toggling between sleep and wake periods, the asynchronous protocol proposed in [18] may force sensors to be awake for longer periods but avoids frequent transitions from sleep to wake periods.

References

1. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

2. S. Bandyopadhyay and E. Coyle. An efficient hierarchical clustering algorithm for wireless sensor networks. In *Proc. IEEE INFOCOM 2003*, San Francisco, CA, April 2003.
3. D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *IEEE Computer*, 37(8):41–49, 2004.
4. D. Culler and W. Hong. Wireless sensor networks. *Communications of the ACM*, 47(6):30–33, 2004.
5. C.C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris. WiseNET: a ultralow power wireless sensor network solution. *IEEE Computer*, 37(8):62–69, 2004.
6. S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal energy-aware clustering in sensor networks. *Sensors*, 2:258–269, 2002.
7. J. Hill, M. Horton, R. Kling, and L. Krishnamurthy. The platforms enabling wireless sensor networks. *Communications of the ACM*, 47(6):41–46, 2004.
8. K. Langendoen and N. Reijers. Distributed localization algorithm. In *Embedded Systems Handbook*, R. Zurawski (Editor), CRC Press, Boca Raton, FL, 2004.
9. K. Martinez, J.K. Hart, and R. Ong. Sensor network applications. *IEEE Computer*, 37(8):50–56, 2004.
10. D. Nicolescu. Positioning in ad-hoc sensor networks. *IEEE Network*, 18(4):24–29, 2004.
11. S. Olariu, A. Waada, L. Wilson, and M. Eltoweissy. Wireless sensor networks leveraging the virtual infrastructure. *IEEE Network*, 18(4):51–56, 2004.
12. P. Saffo. Sensors, the next wave of innovation. *Communications of the ACM*, 40(2):93–97, 1997.
13. M. Sichitiu and C. Veerarithiphan. Simple accurate synchronization for wireless sensor networks. In *Proc. WCNC'03*, 2003.
14. F. Sivrukaya and B. Yener. Time synchronization in sensor networks: a survey. *IEEE Network*, 18(4):45–50, 2004.
15. K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 16–27, October 2000.
16. A. Waada, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones. Training a wireless sensor network. *Mobile Networks and Applications*, 10(1):151–168, 2005.
17. B. Warneke, M. Last, B. Leibowitz, and K. Pister. SmartDust: communicating with a cubic-millimeter computer. *IEEE Computer*, 34(1):44–51, 2001.
18. Q. Xu, R. Ishak, S. Olariu, and S. Salleh. On asynchronous training in sensor networks. In *3rd Intl. Conf. on Advances in Mobile Multimedia*, K.Lumpur, September 2005.
19. V.V. Zhirnov, and D.J.C. Herr. New frontiers: self-assembly and nano-electronics. *IEEE Computer*, 34(1):34–43, 2001.