2008

Efficient Corona Training Protocols for Sensor Networks

Alan A. Bertossi

Stephan Olariu
Old Dominion University

Cristina M. Pinotti

Original Publication Citation

Bertossi, A. A., Olariu, S., & Pinotti, C. M. (2008). Efficient corona training protocols for sensor networks. *Theoretical Computer Science, 402*(1), 2-15. doi:10.1016/j.tcs.2008.03.007

Theoretical Computer Science 402 (2008) 2–15

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Efficient corona training protocols for sensor networks

Alan A. Bertossi [a], Stephan Olariu [b], Cristina M. Pinotti [c,*]

[a] *Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy*

[b] *Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA*

[c] *Department of Mathematics and Computer Science, University of Perugia, Via Vanvitelli 1, 06123 Perugia, Italy*

A R T I C L E   I N F O

Keywords:
Wireless sensor networks
Training
Localization
Network protocols
Algorithms

A B S T R A C T

Phenomenal advances in nano-technology and packaging have made it possible to develop miniaturized low-power devices that integrate sensing, special-purpose computing, and wireless communications capabilities. It is expected that these small devices, referred to as *sensors*, will be mass-produced and deployed, making their production cost negligible. Due to their small form factor and modest non-renewable 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, two scalable energy-efficient training protocols are proposed for massively-deployed sensor networks, where sensors are initially anonymous and unaware of their location. The training protocols are lightweight and simple to implement; they are 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 technological breakthroughs in ultra-high integration and low-power electronics have enabled the development of miniaturized battery-operated sensor nodes (*sensors*, for short) that integrate signal processing and wireless communications capabilities [1,35]. Together with innovative and focused network design techniques that will make possible massive deployment and sustained low power operation, the small size and cost of individual sensors are a key enabling factor for a large number of applications. Indeed, aggregating sensors into sophisticated computational and communication infrastructures, called *wireless sensor networks*, has a significant impact on a wide array of applications ranging from smart kindergarten [16,29], smart learning environments [5,11,20], habitat monitoring [18,31], environment monitoring [14,32], greenhouse and vineyard experiments [6,12], forest fire detection [7], helping the elderly and the disabled [8,14,30], among others. These prototypes provide solid evidence of the usefulness of sensor networks and suggest that the future will be populated by pervasive sensor networks that will redefine the way we live and work [1,5,9].

The peculiar characteristics of sensor networks (a massive deployment of sensors, the anonymity of individual sensors, a limited energy budget per sensor, and a possibly hostile environment) pose unique challenges to the design of protocols. First of all, the limited energy budget requires the design of ultra-lightweight communication protocols. To achieve this goal, how data collected by sensors are queried and accessed and how concurrent sensing can be performed internally are of significance. An important guideline in this direction is to perform as much local data processing at the sensor level as

* Corresponding author.
E-mail address: pinotti@unipg.it (C.M. Pinotti).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2008.03.007

possible, avoiding the transmission of raw data through the sensor network. Second, the sensor networks have to interface to the outside world. The simplest technique involves using one or several *anchor* nodes, deployed alongside with the sensors, each having a full range of computational and communication capabilities. In this scenario, the raw data collected by individual sensors are fused, in stages, and forwarded to the nearest anchor that provides the interface to the outside world. This implies that the sensor network must be multi-hop. Third, the random deployment results in sensors initially unaware of their location and of the network topology. 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 [16,33]. One notable application is that of *clustering*, where the set of sensors deployed in an area is partitioned into clusters [1,2,10,29], each corresponding to a small region of indistinguishable sensors. 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 location.

The task of determining an exact geographic location is referred to as *localization* and has been extensively studied in the literature (see e.g. [13,22] for surveys). The immediate approach to provide the exact geographic position to sensors is obviously based on GPS. Such an approach, however, is unsuitable for low-cost and small-sized sensors because GPS requires an extensive infrastructure (i.e. satellites). To reduce the infrastructure complexity and the sensor dependence on special hardware, prominent solutions assume the existence of several anchor nodes, that are aware of their location because they are the only GPS-equipped. Most solutions are distributed, that is they do not require centralized computation, and rely on each sensor determining its location with only limited communication with nearby sensors [15,21,23]. In general, a distributed protocol may follow up to three phases for determining the individual sensor positions [13]. First, the distances between sensors and anchor nodes are determined by flooding information into the network starting from the anchor nodes (e.g., by counting the number of hops). Then, when each sensor has located enough anchors in its neighborhood, it derives its position from the distances and the positions of its neighbour anchors (e.g., by applying multilateration or multiangulation techniques). Finally, the sensor position can be further refined by using information about the range to, and the position of, neighboring sensors. The main disadvantage of such distributed protocols is the fact that they may incur too much communication overhead and a large energy consumption due to the lack of synchronization.

Instead, the task of acquiring a coarse-grain location is referred to as *training*. Such a task has been considered in several recent papers [3,16,33,34] and it is also the topic of the present paper. The main characteristic of the training protocols studied so far relies on using a single anchor node, called a *sink*, which has a steady power supply and can send long range directional broadcasts to the sensors so as to impose a coordinate system on the area it covers. The process is centralized because it uses only asymmetric broadcasts (from the sink to the sensors) without multihop communications among the sensors. On the other hand the sensors, which act with the intent of being localized, cooperate by using the received information to deduce their coarse-grain location. Summarizing, using the taxonomy in [22], such a training approach has cooperative targets and active infrastructure. Such an approach tends to be more efficient and more effective than distributed localization because it allows explicit synchronization thus optimizing the protocol designs to achieve better performance of both the sink and the sensors. Moreover, with respect to the 3-phase approach outlined in [13], training corresponds to the first phase, which computes the distances from the anchor to the sensors, but it is performed in a fully centralized manner instead of being completely distributed. Note that, by asymmetric broadcasts, sensors learn unidirectional distances from the sink to them. Such distances do not necessarily coincide with the reverse multihop distances from them to the sink, which instead depend on the network connectivity.

The detailed contribution of the present paper is to exhibit two new training protocols which combine the single centralized phase of [33] with a second distributed phase which proceeds incrementally, starting from a core set of sensors having learned their location in the first centralized phase and adding new groups of sensors which derive their location by hop counting. The main advantage consists in lowering the overall time for training (from a linear to a square-root function) still maintaining the same sensor energy consumption of the single phase, fully centralized protocol in [33]. Moreover, the second protocol can achieve an optimal, constant number of sensor wake/sleep transitions still maintaining the same overall time for training. In addition it shows that the training protocols can benefit from longer sleep periods because they get synchronization constraints less stringent. The remainder of this paper is organized as follows. Section 2 discusses the network model used throughout the work and introduces the task of training. 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 [33]. Section 3 is the backbone of the entire paper, presenting the theoretical underpinnings of the first training protocol. In the 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 of as a visit of a complete binary tree, whose leaves represent coronas, whose node preorder numbers are related to the time slots, and whose node 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. Section 4 shows how the first protocol can be extended so as to reduce the number of sensor wake/sleep transitions. The computation of the first phase can be thought of as a breadth-first-search of a complete q-ary tree, whose nodes are numbered by increasing levels and, at the same level, from left to right.

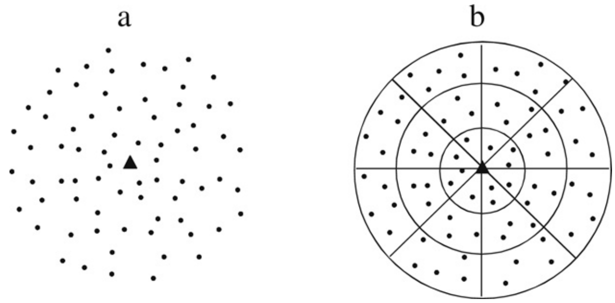

Fig. 1. (a) A sensor network with a central sink. (b) The trained sensor network.

Section 5 evaluates the energy drained by both training protocols under a realistic estimate of the power consumed by the sensors in their different operative modes. Finally, Section 6 offers concluding remarks.

2. The network model

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 R as illustrated in Fig. 1(a). For simplicity, we assume that the sink is centrally placed, although this is not really necessary.

We assume sensors to be devices that possess three basic capabilities – sensory, computation, and wireless communication – and 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, the 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. No sensor has global information about the network;
f. 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 years may need to scavenge energy from the ambient environment, e.g. from vibrations, kinetics, magnetic fields, seismic tremors, pressure, etc. [19,24,25].

We assume that the task of training imposes a coordinate system onto the sensor network. Such a coordinate system involves establishing [33]:

1. *Coronas*: The deployment area is covered by k coronas determined by k concentric circles, centered at the sink, whose radii are $0 < r_1 < r_2 < \cdots < r_k = R$.
2. *Wedges*: The deployment area is ruled into a number of equiangular wedges, centered at the sink, which are established by directional transmission [16].

As illustrated in Fig. 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.

We assume that the number k of coronas is a power of two and is known to the sensors. We assume also that each sensor has the same transmission range r and that the radius r_i is equal to ir, namely the corona width is $r_{i+1} - r_i = r$. Recently, Olariu and Stojmenovic [17] showed that there are other choices for the widths of the coronas that promote extended longevity of the network. However, we shall not embark on this topic here.

Finally, it is assumed that the time is ruled into slots and that the sensors can synchronize to the master clock running at the sink [27,28].

3. The corona training protocol

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

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. The corona training protocol consists of two phases: a first centralized sink-driven phase, during which the sensors learn the

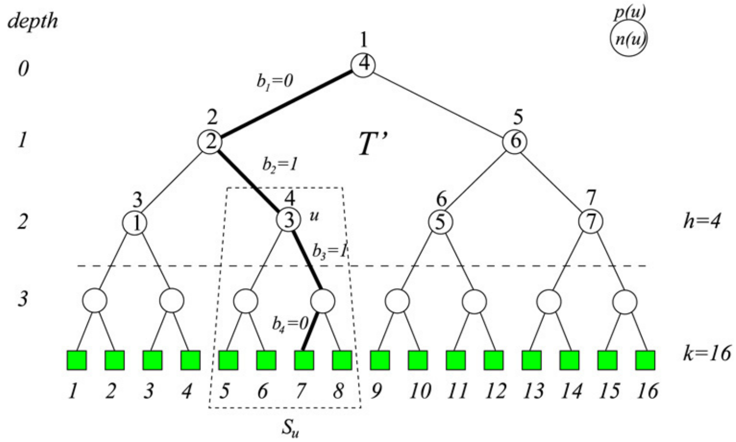

Fig. 2. Illustrating the corona training protocol. Labels outside nodes of T' give their preorder numbers, while labels inside give their inorder numbers. Leaves represent coronas, numbered from 1 to k.

leftmost bits of the corona to which they belong, followed by a second distributed phase, where sensors learn the remaining bits.

To see how the first phase is done, it is useful to view the protocol computation as a visit of a complete binary tree, whose leaves represent coronas, whose node preorder numbers are related to time slots when the sensors wake up, and whose node inorder numbers are related to the transmission range used by the sink.

3.1. Binary tree representation

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, $(1 \le \ell \le k)$, be an arbitrary leaf and let $b_1, b_2, \dots, b_{\log k}$ 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 Fig. 2, where $k = 16$. By applying (1) to leaf 7, we obtain: $7 = 1 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$.

Let h be an integer known to the sensors which is a power of two such that $1 \le h \le k/2$. Consider the subtree T' consisting of the uppermost $2h - 1$ nodes of T. Refer again to Fig. 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 $b_1, b_2, \dots, b_i$ be the edge labels on the unique path from the root to u, where i is the depth of u in T' and $1 \le i \le \log h$. Obviously, 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 results.

Lemma 3.1. *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 [33] and is reported in the Appendix for the sake of completeness. □

Lemma 3.2. *Let u be an arbitrary node of depth i in T'. Then, the inorder number $n(u)$ of u in T' 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. See the Appendix. □

As an example, consider node u in Fig. 2. Applying Lemmas 3.1 and 3.2, one gets $p(u) = 1 + 1 + \frac{4}{2^1} = 4$ and $n(u) = 4 - \frac{4}{2^1} + \frac{4}{2^2} = 3$.

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.

3.2. First centralized phase

We now return to the details of the corona training protocol. Recall that the goal of the 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 first phase is sink-driven and lasts for $2h - 1$ time slots. During this phase, the sensors learn the leftmost $\log h + 1$ bits of the corona to which they belong. 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 and the awake sensors in the centralized phase are described as follows.

In time slot s_1, all the sensors are awake and the sink transmits at 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 $b_1 = 0$, while the others set $b_1 = 1$. In general, referring to the T' tree, consider a generic time slot s_z, with $1 \le z \le 2h - 1$. Let u be the node of T' whose preorder number, $p(u)$, satisfies $p(u) = z$, and let S_u be the subtree of T rooted at u, as illustrated in Fig. 2. At time slot s_z, the sink 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 $i \le \log h$, it can easily compute the value c_{i+1} given in Lemma 3.1, 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 $s_{2h+2\gamma}$ for executing the second phase of the protocol.

3.3. Correctness

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

Lemma 3.3. *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. $\square$

As an example, refer again to Fig. 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 3.4. *Consider a time slot s_z, with $1 \le z \le 2h-1$. At time slot s_z, all the sensors belonging to any corona c, with $1 \le c \le k$, 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 $p(u) \le z$.*

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 $p(u) \le 1$. 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 of 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 3.3, 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 3.3, 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 3.4, refer again to node u of Fig. 2. Only the sensors in the leaves of S_u are awake in time slot $s_{p(u)} = s_4$, while the sink 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 3.5. *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.*

3.4. Second distributed phase

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

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

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

3.5. Complexity analysis

In order to analytically evaluate the complexity of the training protocol, the following notations are introduced. Let ν be the number of wake/sleep transitions required by a sensor to be trained in the worst case. Moreover, let ω be the overall sensor awake time and τ be the total time for training.

In the protocol, 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 slot $s_{(2h+2(2h-1)+1)+\frac{k}{2h}-1}$. Therefore, the entire corona training protocol finishes at time $\tau = 6h - 2 + \frac{k}{2h}$, and thus $\tau = O(h + \frac{k}{h})$. The total time τ is minimized when $h = \Theta(\sqrt{k})$, and in such a case it becomes $O(\sqrt{k})$, improving over the $O(k)$ time of the training protocol presented in [33].

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, thus avoiding repeated wake/sleep transitions that are known to waste energy. To see this, observe that the sensors remain awake $\log h$ time slots 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 at most $2 + \frac{k}{h}$ time slots during the second phase. Therefore, the sensor awake time ω is at most $\log h + \frac{k}{h} + 2$, which is minimized when $h = \Theta(\frac{k}{\log k})$. In such a case $\omega = O(\log k)$, 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 ν of wake/sleep transitions is at most $\log h + 3$. Precisely, referring again to Fig. 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, with a minimum gap of two between u and its parent when u is at depth $\log h$. Thus, the worst case arises for the sensors in the coronas of

Table 1
Complexity of the corona training protocol

Performance measure	Complexity	$h = \Theta(\sqrt{k})$	$h = \Theta\left(\frac{k}{\log k}\right)$
Overall time τ	$O\left(h + \frac{k}{h}\right)$	$O(\sqrt{k})$	$O\left(\frac{k}{\log k}\right)$
Sensor awake time ω	$O\left(\log h + \frac{k}{h}\right)$	$O(\sqrt{k})$	$O(\log k)$
♮ Wake/Sleep transitions ν	$O(\log h)$	$O(\log k)$	$O\left(\log\left(\frac{k}{\log k}\right)\right)$

The sensor awake time ω is optimal when $h = \Theta(\frac{k}{\log k})$, while the overall time τ is minimized when $h = \Theta(\sqrt{k})$.

group $2h - 1$, which go through exactly $\log h + 1$ transitions during the first phase, plus two more transitions during the second phase. The complexity achieved by the corona training protocol for the above mentioned performance measures is summarized in Table 1.

4. The extended protocol

The training protocol discussed in the previous section optimizes the sensor awake time ω by allowing the sensors to quickly toggle between sleep and wake periods. Depending on the corona to which they belong, the sensors may either wake up or go to sleep for just a single time slot. Hence, a single time slot must be sufficiently long to allow both radio startup and shutdown, which together consume a not negligible amount of time.

In order to compensate for the time wasted in such repeated wake/sleep transitions, one can prolong the wake and sleep periods to a fixed amount of time slots. Precisely, the previous protocol can be extended so that the sensors still toggle between sleep and wake periods depending on the protocol computation, but when sensors wake up or go to sleep they do not change mode for at least q slots. Being awake for q time slots, in the first sink-driven phase of the new protocol sensors learn additional $\log q$ bits at a time. At the end of the first phase, sensors will have learned the leftmost $b_1, b_2, \ldots, b_{d \log q}$ bits of their corona number, where d is the number of wake periods experienced by the sensors. Then, the second distributed phase can proceed as in the previous protocol.

4.1. *q-ary tree representation*

To illustrate the extended corona training protocol, consider again a k-leaf tree T, whose leaves are numbered left to right from 1 to k, where k is a power of two. Let q and m be two integers, known to the sensors, such that q is a power of two and m is a power of q with $m \leq k$. Assume that the subtree T', consisting of the uppermost $\log_q m + 1$ levels of T, is a complete q-ary tree. Note that, since k is a power of two, T results to be a complete tree. As an example, refer to Fig. 3, where $k = 128$, $q = 4$, $m = 16$, and T' consists of the uppermost $\log_4 16 + 1 = 3$ levels. Note that there are q^i nodes at depth i in T', while there are $\sum_{h=0}^{i} q^h = \frac{q^{i+1}-1}{q-1}$ nodes with depth at most i, where $0 \leq i \leq \log_q m$.

The edges of T' are labeled by binary strings of length $\log q$ in such a way that an edge leaving from a node to its j-th child is labeled by the binary representation of $j - 1$, with $1 \leq j \leq q$, as shown in Fig. 3. In particular, the binary string labeling the edges leaving from a node u at depth i towards its children consists of the bits indexed $b_{i \log q+1}, \ldots, b_{(i+1) \log q}$. It is easy

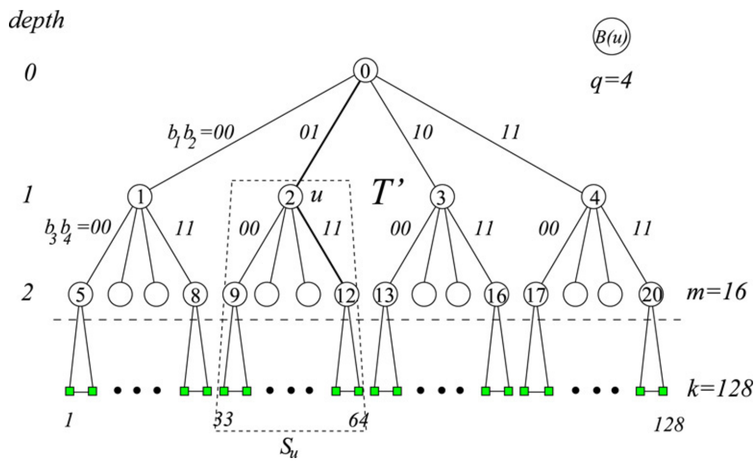

Fig. 3. Illustrating the extended corona training protocol. Labels inside nodes of T' give their bfs-order numbers. Leaves represent coronas, numbered from 1 to k.

to check that the edge labels $b_1, b_2, \ldots, b_{i \log q}$ of the unique path leading from the root to u gives the binary representation of the relative position $pos(u)$ of node u among all the nodes at depth i, namely

$$pos(u) = \sum_{h=1}^{i \log q} b_h 2^{i \log q - h} \tag{2}$$

where such a relative position is counted from left to right, starting from 0. Since the root r has no incoming edges, its position $pos(r)$ is defined to be 0.

Let the nodes of T' be numbered according to their breadth-first-search order, starting with 0 at the root. Precisely, the nodes are numbered by increasing levels, and at the same level, from left to right. It is easy to check by an inductive argument that the bfs-order number $B(u)$ of any node u in T' is

$$B(u) = \begin{cases} 0 & \text{if } u \text{ is the root} \\ qB(v) + j & \text{if } u \text{ is the } j\text{-th child of } v, \text{ with } 1 \le j \le q. \end{cases} \tag{3}$$

One can see that a node u with bfs-order number $B(u)$ is at depth $i = \lceil \log_q((q-1)B(u) + 2) \rceil - 1$ because the bfs-order numbers of the nodes at depth i range from $\frac{q^i - 1}{q-1}$ to $\frac{q^{i+1}-1}{q-1} - 1$. The position $pos(u)$ can be derived from $B(u)$ and i as $pos(u) = B(u) - \frac{q^i - 1}{q-1}$. Moreover, the bfs-order number $B(v)$ of the parent v of u is $\lfloor \frac{B(u)-1}{q} \rfloor$. One can see that u is the j-th child of v, where $j = (B(u) - 1) \bmod q + 1$, and that $pos(u) = q\, pos(v) + j - 1$. For example, refer to node u in Fig. 3. Since $B(u) = 2$, the depth of u is $i = \lceil \log_4(3 * 2 + 2) \rceil - 1 = 1$, its position is $pos(u) = B(u) - \frac{q^i - 1}{q-1} = 2 - \frac{4^1 - 1}{3} = 1$, its parent v has $B(v) = \lfloor \frac{2-1}{4} \rfloor = 0$, u is the $((2-1) \bmod 4 + 1)$-th child of v, and $pos(u) = q\, pos(v) + j - 1 = 0 * 4 + 2 - 1 = 1$.

Lemma 4.1. *Let u be an arbitrary node, other than the root, of depth i in T'. Then, the bfs-order number $B(u)$ of u is given by*

$$B(u) = \sum_{d=1}^{i} 2^{(i-d)\log q} \left(\sum_{h=1}^{\log q} b_{(d-1)\log q + h} 2^{\log q - h} + 1 \right).$$

Proof. See the Appendix. □

We can now turn our attention to the description of the first phase of the extended protocol. In our setting the bfs-order numbers are related to the time slots in the training protocol as well as to the sink transmission ranges.

4.2. First phase

In time slot s_1 all the sensors are awake and they will remain awake for q time slots. During such q slots, the sink transmits with increasing power levels corresponding to $r_{\frac{k}{q}}, r_{2\frac{k}{q}}, \ldots, r_{(q-1)\frac{k}{q}}, r_k$, respectively, and each sensor counts how many times it hears the sink transmission. In time slot s_q, a sensor that has received σ messages from the sink sets its local bits $b_1, \ldots, b_{\log q}$ to the binary representation of $q - \sigma$. Note that the sensors that have heard σ signals are those belonging to the coronas corresponding to the leaves of the subtree S_u rooted at node u such that $B(u) = q - \sigma + 1$. At the end of time slot s_q such sensors go to sleep and they will wake up again at time slot $s_{q(q-\sigma+1)+1}$. As an example, referring again to Fig. 3, the sensors that learned $b_1 b_2 = 01$ correspond to those in the leaves of the subtree S_u where $B(u) = 4 - 3 + 1 = 2$ and they will wake up at $s_{4*2+1} = s_9$.

In general, the procedure performed by the sink in the centralized phase is described as follows. Consider a generic time slot s_z, with $1 \le z \le \frac{q^{\log_q m + 1} - 1}{q-1} - 1 = \frac{q(m-1)}{q-1}$. Let u be the node of T' such that its bfs-order number is $B(u) = z$. At time slot s_z, the sink transmits with a power level equal to $r_{(pos(u)+1)\frac{k}{q^i}}$, where $i = \lceil \log_q((q-1)z + 2) \rceil - 1$ is the depth of u and $pos(u) = z - \frac{q^i - 1}{q-1}$ is its relative position at level i. While the sink is transmitting at time slot s_z, the awake sensors must be those belonging to the coronas which are the leaves of the subtree S_v rooted at the parent v of u, whose bfs-order number is $B(v) = \lfloor \frac{z-1}{q} \rfloor$.

By contrast, the procedure performed by a sensor is the following. At time slot s_1 all the sensors wake up for the first time and set $i = 1$, which counts how many times a sensor woke up so far. In general, a sensor wakes up at a time slot s_z, where z is such that $z = tq + 1$ and $0 \le t \le \frac{m-1}{q-1} - 1$, increments its counter i, and stays awake up to s_{z+q-1}. If the sensor hears σ sink transmissions during such q time slots, at the end of time slot s_{z+q-1} it sets its bits $b_{(i-1)\log q + 1}, \ldots, b_{i \log q}$ to the binary representation of $q - \sigma$. Then, if $i < \log_q m$, it computes the value $q(z + q - \sigma) + 1$, which the sensor can easily derive from all the bits learned so far, as proved in Lemma 4.1. Thus, the sensor goes to sleep and will wake up again at time slot $s_{q(z+q-\sigma)+1}$. If $i = \log_q m$, then the sensor will wake up again at time slot $s_{\frac{q(m-1)}{q-1}+1}$ for the second phase.

4.3. Correctness

In order to verify the correctness of the first phase of the extended corona training protocol, the following lemmas are useful.

Lemma 4.2. *Let u be any node of T' at depth $i > 0$, and let v be the parent of u. The subtree S_v rooted at v contains $\frac{k}{q^{i-1}}$ leaves, whose indices range from $pos(v)\frac{k}{q^{i-1}} + 1$ to $(pos(v)+1)\frac{k}{q^{i-1}}$.*

Proof. The proof is by induction on the bfs-order number $B(u)$. As the basis, observe that if $B(u) = 1$, then u is at depth 1 and its parent v is the root. Since S_v contains all the leaves of T and $pos(v) = 0$, the indices of the leaves range from 1 to $\frac{k}{q^0} = k$. For the inductive step, assume the statement true up to node w such that $B(w) = B(u) - 1$. Three cases may arise. If w and u share the same parent, then S_v is the same and the proof follows by inductive hypothesis. If w and u do not share the same parent but they are at the same depth, let v' be the parent of w. Since $pos(v) = pos(v') + 1$, the subtree S_v contains the same number of leaves as $S_{v'}$, whose indices range from $(pos(v')+1)\frac{k}{q^{i-1}} + 1 = pos(v)\frac{k}{q^{i-1}} + 1$ up to $(pos(v')+1)\frac{k}{q^{i-1}} + \frac{k}{q^{i-1}} = (pos(v')+2)\frac{k}{q^{i-1}} = (pos(v)+1)\frac{k}{q^{i-1}}$. Finally, if u is one level deeper than w, let v' be the grand-parent of u. The subtree rooted at v' has size $\frac{k}{q^{i-2}}$, and $pos(v') = pos(v) = 0$. Hence, the subtree S_v rooted at v contains $\frac{1}{q}\frac{k}{q^{i-2}} = \frac{k}{q^{i-1}}$ leaves, whose indices range from $pos(v')\frac{k}{q^{i-2}} + 1 = pos(v)\frac{k}{q^{i-2}} + 1 = 1$ to $pos(v')\frac{k}{q^{i-2}} + \frac{k}{q^{i-1}} = \frac{k}{q^{i-1}} = (pos(v)+1)\frac{k}{q^{i-1}}$. $\square$

Lemma 4.3. *Let u be any node of T' at depth $i > 0$, and let v be the parent of u. At time slot $s_z = s_{B(u)}$, the sink transmission reaches all the coronas indexed up to $pos(v)\frac{k}{q^{i-1}} + ((B(u)-1) \bmod q + 1)\frac{k}{q^i}$.*

Proof. At time slot $s_z = s_{B(u)}$, the sink transmits with a power level equal to $r_{(pos(u)+1)\frac{k}{q^i}}$. Since $pos(u) = q\,pos(v) + j - 1$ if u is the j-th child of v, with $1 \le j \le q$, and $j = (B(u)-1) \bmod q + 1$, one has $(pos(u)+1)\frac{k}{q^i} = (q\,pos(v)+j)\frac{k}{q^i} = pos(v)\frac{k}{q^{i-1}} + ((B(u)-1) \bmod q + 1)\frac{k}{q^i}$, as stated. $\square$

Corollary 4.4. *Consider the q time slots $s_{q\lfloor\frac{z-1}{q}\rfloor+1}, \ldots, s_{q\lfloor\frac{z-1}{q}\rfloor+q}$, with $1 \le z \le \frac{q(m-1)}{q-1}$. During such q time slots, the sink has reached $q - j + 1$ times the coronas indexed $pos(v)\frac{k}{q^{i-1}} + 1, \ldots, pos(v)\frac{k}{q^{i-1}} + ((j-1) \bmod q + 1)\frac{k}{q^i}$, where v is the node at depth $i-1$ such that $B(v) = \lfloor\frac{z-1}{q}\rfloor$ and $1 \le j \le q$.*

Proof. The proof follows from Lemmas 4.2 and 4.3. $\square$

As an example, refer again to Fig. 3, and consider the node u having $B(u) = 13$, which is at depth $i = 2$ and whose parent v has $B(v) = 3$ and $pos(v) = 2$. The subtree S_v contains 32 leaves, indexed from 65 to 96. At time slot s_{13}, the sink transmission reaches all the coronas indexed up to $2\frac{128}{4} + (12 \bmod 4 + 1)\frac{128}{4^2} = 72$. During the 4 time slots $s_{13}, \ldots, s_{16}$, the sink has reached 4 times the coronas indexed from $2\frac{128}{4} + 1 = 65$ to $2\frac{128}{4} + \frac{128}{4^2} = 72$, 3 times the coronas from 73 to 80, twice those from 81 to 88, and once from 89 up to 96.

Lemma 4.5. *Consider the q time slots $s_{q\lfloor\frac{z-1}{q}\rfloor+1}, \ldots, s_{q\lfloor\frac{z-1}{q}\rfloor+q}$, with $1 \le z \le \frac{q(m-1)}{q-1}$. Only the sensors that belong to the coronas which are the leaves of the subtree S_v such that $B(v) = \lfloor\frac{z-1}{q}\rfloor$ stay awake during such q slots.*

Proof. The proof is by induction on z. As the basis observe that, according to the protocol computation, all the sensors wake up at time slot s_1, stay awake until s_q, and then go to sleep. Therefore, those sensors belong to the coronas which are the leaves of the subtree rooted at the root, which has $B(v) = \lfloor\frac{z-1}{q}\rfloor = 0$. For the inductive step, assume the statement true for $z-1$. If $z \ne tq + 1$, then the awake sensors in time slot s_z remain the same as in s_{z-1}. Indeed $\lfloor\frac{z-2}{q}\rfloor = \lfloor\frac{z-1}{q}\rfloor$, and hence v remains the same. Otherwise, if $z = tq + 1$ with $t \ge 1$, by inductive hypothesis, the sensors belonging to the coronas which are the leaves of S_v such that $B(v) = \lfloor\frac{z-2}{q}\rfloor = t - 1$ were awake for q time slots and they are no longer awake at time slot s_z because they went to sleep at the end of s_{z-1}. The sensors that wake up at $s_z = s_{tq+1}$ are those that belong to the subtree S_v for which $B(v) = t$. Since $z = tq + 1$, it holds $t = \lfloor\frac{z-1}{q}\rfloor$ as claimed. $\square$

Theorem 4.6. *Consider a time slot s_z, with $1 \le z \le \frac{q(m-1)}{q-1}$. At time slot s_z, all the sensors belonging to any corona c, with $1 \le c \le k$, have learned bits $b_1, b_2, \ldots, b_{i\log q}$, where i is the depth of the deepest node u on the unique path from the root to leaf c such that $B(u) \le q\lfloor\frac{z}{q}\rfloor$.*

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 $B(u) \le \lfloor\frac{1}{q}\rfloor = 0$. Observe that the depth of the root u is 0, all the sensors are awake, and they have learned no bits so far. Therefore, all the sensors in any corona c have learned an empty string of bits.

For the inductive step, assume the statement true for $z-1$. At time slot s_z, if $\lfloor\frac{z-1}{q}\rfloor = \lfloor\frac{z}{q}\rfloor$, then the deepest node u on the unique path from the root to any leaf c such that $B(u) \le q\lfloor\frac{z}{q}\rfloor$ is unchanged, and the claim trivially follows. Otherwise, that is when $\lfloor\frac{z}{q}\rfloor = \lfloor\frac{z-1}{q}\rfloor + 1$, the deepest node u has changed for all the sensors that belong to the coronas which are the leaves of the subtree S_v such that $B(v) = \lfloor\frac{z-1}{q}\rfloor$. Specifically, the old deepest node v has been replaced by its q children at

Table 2
Complexity of the extended corona training protocol

Performance measure	Complexity	$q = m = \Theta(\sqrt{k})$	$q = O(1), m = \Theta\left(\frac{k}{\log k}\right)$
Overall time τ	$O\left(m + \frac{k}{m}\right)$	$O(\sqrt{k})$	$O\left(\frac{k}{\log k}\right)$
Sensor awake time ω	$O\left(q \log_q m + \frac{k}{m}\right)$	$O(\sqrt{k})$	$O(\log k)$
♮ Wake/Sleep transitions ν	$O(\log_q m)$	$O(1)$	$O\left(\log\left(\frac{k}{\log k}\right)\right)$

When $q = m = \Theta(\sqrt{k})$, the number ν of wake/sleep transitions is optimal and the overall time τ is minimized. When $q = O(1)$, the extended protocol matches the complexity of the previous protocol.

depth i having bfs-order numbers spanning from $z - q + 1 = q\lfloor \frac{z-1}{q} \rfloor + 1$ to $z = q\lfloor \frac{z-1}{q} \rfloor + q$. By Lemma 4.5 and Corollary 4.4, the sensors that belong to the coronas which are the leaves of the j-th child of v have heard $q - j + 1$ sink transmissions, and they have set their bits $b_{(i-1)\log q+1}, \ldots, b_{i \log q}$ to the binary representation of $j - 1$. Hence, such sensors have learned bits $b_1, b_2, \ldots, b_{i \log q}$. $\square$

Corollary 4.7. *At time slot $s_{\frac{q(m-1)}{q-1}}$, the first phase of the extended corona training protocol is completed, and the sensors belonging to any corona c, with $1 \le c \le k$, have learned the leftmost $\log m$ bits, $b_1, \ldots, b_{\log m}$, of the binary representation of their corona number minus one.*

4.4. Complexity analysis

In order to evaluate the complexity of the extended corona training protocol, recall that the second distributed phase behaves as in the previous protocol, except that now there are m groups, each of $\frac{k}{m}$ consecutive coronas, and that all the sensors learn the remaining $\log k - \log m$ bits, $b_{\log m+1}, \ldots, b_{\log k}$. Such a phase starts at time slot $s_{\frac{q(m-1)}{q-1}+1} = s_{\frac{qm-1}{q-1}}$ and lasts for $2m + \frac{k}{m}$ time slots. Since the first phase takes $\frac{q(m-1)}{q-1}$ time slots, the overall time τ of the extended corona training protocol is $\frac{q(m-1)}{q-1} + 2m + \frac{k}{m}$. Moreover, the number ν of wake/sleep transitions is $\log_q m + 2$ because sensors wake up once for each level of T' during the first phase, and just twice during the second phase. As regard to the sensor awake time ω, observe that the sensors remain awake $q \log_q m$ time slots during the first phase, because they wake up for q time slots once for each level of T', and additional $2 + \frac{k}{m}$ time slots during the second phase.

Note that the overall time τ is minimized when $m = O(\sqrt{k})$, and in such a case it becomes $O(\sqrt{k})$. Similarly, when $q = \Theta(m)$, the number ν of transitions is minimized and becomes $O(1)$, which is clearly optimal. Therefore, choosing $m = q = \Theta(\sqrt{k})$, the extended protocol maintains the same $O(\sqrt{k})$ overall time and $O(\sqrt{k})$ sensor awake time as the protocol presented in Section 3 and in addition it achieves an optimal number of wake/sleep transitions. It is worth noting that the extended protocol has the same complexity as that presented in Section 3 when one chooses $q = O(1)$ and $m = \Theta(\frac{k}{\log k})$. In summary, the complexity achieved by the extended protocol is illustrated in Table 2.

5. Energy consumption

In this section, the energy drained by both training protocols is evaluated under a realistic estimate of the power consumed by the sensors in their different operative modes.

During the training task, when a sensor is awake, its CPU is active and its radio is listening, receiving, or transmitting. Instead, when a sensor is sleeping, its CPU is not active, its timer is on, and its radio is off. Let e_{awake}, e_{TX}, and e_{sleep} be the energy consumed during a time slot by a sensor when it is listening/receiving, transmitting, or sleeping, respectively. Since the radio startup and shutdown require a not negligible overhead, let e_{trans} denote the energy consumed for a sleep/wake transition followed by a wake/sleep transition. Recalling that ν, ω, and τ denote the number of transitions, the overall sensor awake time, and the total time for training, respectively, and observing that a sensor transmits only once during the whole training process, the total energy E depleted by a sensor can be upper bounded as:

$$E \le \nu e_{\text{trans}} + (\omega - 1)e_{\text{awake}} + e_{\text{TX}} + (\tau - \omega)e_{\text{sleep}}. \tag{4}$$

In particular, the energy drained by the training protocol of Section 3 is obtained from Eq. (4) by substituting the upper bounds for ν, ω, and τ given in Section 3.5, thus having:

$$E \le (\log h + 3)e_{\text{trans}} + \left(\log h + \frac{k}{2h} + 1\right) e_{\text{awake}} + e_{\text{TX}} + (6h - \log h - 4)e_{\text{sleep}}. \tag{5}$$

Similarly, the energy spent by the extended protocol of Section 4 is derived from Eq. (4) by using the upper bounds provided in Section 4.4:

$$E \le (\log_q m + 2)e_{\text{trans}} + \left(q \log_q m + \frac{k}{m} + 1\right) e_{\text{awake}} + e_{\text{TX}} + \left(\frac{q(m-1)}{q-1} + 2(m-1) - q \log_q m\right) e_{\text{sleep}}. \tag{6}$$

Table 3
Estimate of sensor power consumption in different operational modes at 2.5 V and with a sensor transmission range r of 10 m

Sensor mode	Current draw	Power consume
CPU inactive, timer on, radio off	6 μA	0.015 mW
CPU switch on, radio startup	3 mA	15 mW
CPU switch off, radio shutdown	3 mA	15 mW
CPU active, radio listening or RX	12 mA	32 mW
CPU active, radio TX	20 mA	50 mW

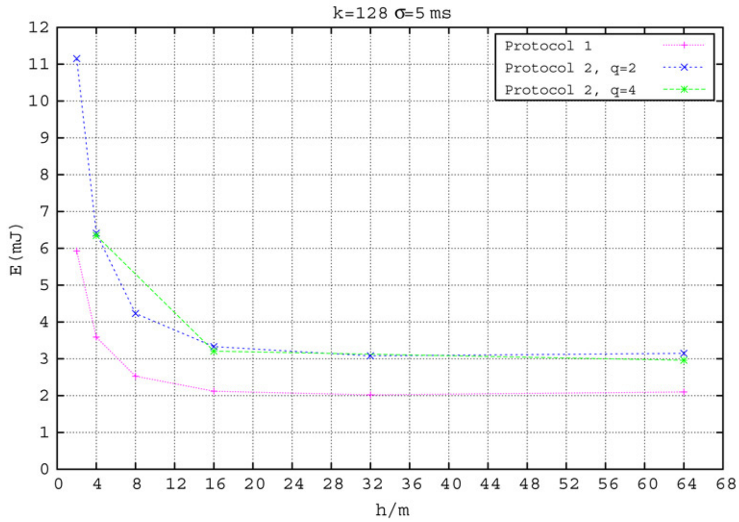

Fig. 4. Energy consumed by the two protocols when $k = 128$ and h and m vary between 2 and 64.

In order to evaluate the energy drained in a realistic setting, Table 3 reports the power consumed by a sensor in different operational modes. The data refer to the TinyNode 584, produced by Shockfish S.A., and are the customary values for the smallest sensors one can buy [4,26]. The sensors operate using 4 dB m transmission power, and hence attaining a transmission range of ten meters, at a bandwidth of 75 kbit/s. The sensors have as a power source two customary 1.2 V batteries, with a capacity of 1900 mA h each, and hence they have an energy supply of 4.56 J. As one can check in the table, listening is nearly as expensive as receiving, while transmitting is the most expensive mode. Although the radio startup and shutdown require a modest power consumption, intermediate between the sleep and active modes, they require a not negligible amount of time (about 2 ms each). This constraint influences the behaviour of the protocols because it gives a lower bound on the time slot length. Indeed, since both protocols alternate sleep and awake periods, the length of their sleep period must be sufficient to allow both radio startup and shutdown, and thus cannot be shorter than 4 ms. In particular, the protocol of Section 3 has sleep periods of just a single time slot, and hence the time slot length, say σ, cannot be shorter than 4 ms. In contrast, the extended protocol of Section 4 has sleep periods of at least q time slots, and thus $q\sigma$ must be no shorter than 4 ms. For instance, if $q = 4$, then $\sigma = 1$ ms is enough. Note that such an amount of time is sufficient to transmit up to 75 bits, much more than the single bit required by the proposed training protocols.

From the data of Table 3, one has that $e_{\text{trans}} = 15 * 2 + 15 * 2 = 60$ mJ, while $e_{\text{sleep}} = 0.015 * \sigma$ mJ, $e_{\text{awake}} = 32 * \sigma$ mJ, and $e_{\text{TX}} = 50 * \sigma$ mJ. Fig. 4 plots the energy consumed by the two protocols when $k = 128$ and $\sigma = 5$ ms. Precisely, Eq. (5) measuring the energy consumed by the protocol of Section 3 (Protocol 1) is evaluated for h assuming as values all the powers of two in the range between 2 and 64. Moreover, Eq. (6), which gives the energy drained by the extended protocol of Section 4 (Protocol 2), is also reported when $q = 2$ and $q = 4$ with m assuming as values all the powers of q between 2 and 64. As one can check in Fig. 4, Protocol 1 consumes less energy than Protocol 2. Observe that a sensor consumes more energy when it is awake for a single time slot than when performs a wake/sleep transition. Therefore, Protocol 2, which reduces the number of transitions slightly increasing the sensor awake time, always loses energy with respect to Protocol 1.

Fig. 5 plots the energy consumed by the two protocols, again for $k = 128$, for the minimum slot length σ allowed. Precisely, the energy consumed by Protocol 1 (Eq. (5)) is reported when $\sigma = 4$, with h ranging between 2 and 64. In addition, the energy required by Protocol 2 (Eq. (6)) is given when $q = 2$, and hence $\sigma = 2$, and also when $q = 4$ and thus $\sigma = 1$. As before, the values of m are the powers of q between 2 and 64. As one can verify in Fig. 5, Protocol 2 is now advantageous because it allows a shorter slot length σ than Protocol 1.

In summary, observing both Figs. 4 and 5, one notes that a suitable choice of h, m, and σ leads to on overall energy depletion of at most 2 mJ. Since the energy supply of a sensor is 4.56 J, the whole training process consumes about 1/2300 of the entire energy budget.

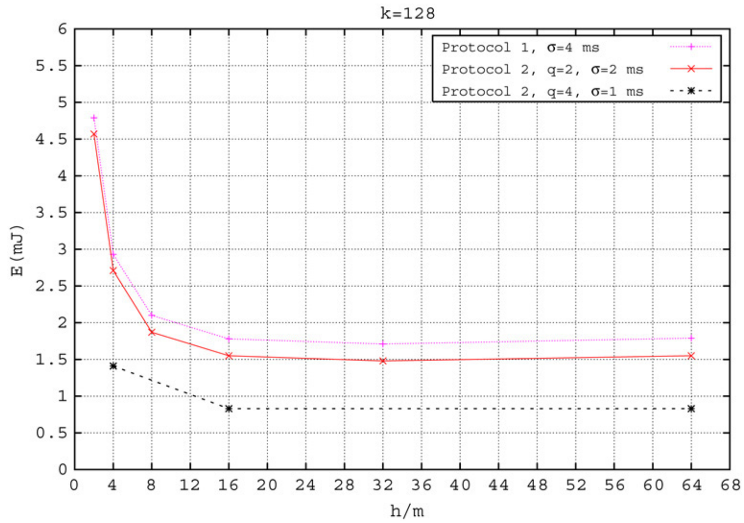

Fig. 5. Energy consumed by the two protocols for minimal time slot lengths.

6. Concluding remarks

In this work new training protocols have been proposed which outperform that originally presented in [33] 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. In particular, the extended protocol of Section 4 allows an optimal, constant number of sensor wake/sleep transitions to be achieved, still maintaining the same overall time for training as the first protocol. Such an extension can also reach an optimal sensor awake time at the cost of a longer overall time and a higher number of sensor wake/sleep transitions, matching the performance of the protocol presented in Section 3. Moreover, the extended protocol, having longer sleep periods than the previous one, allows a shorter time slot length to be adopted thus reducing the total time for training as well as the overall energy depleted by each sensor.

However, several questions still remain open. In particular, the practical behaviour of the proposed protocols could be experimentally checked in an actual sensor network. Indeed, the triggering of the radio range is not so exact and isotropic in the real world as assumed in the present paper. Therefore, one could measure how much the accuracy of the proposed solution decreases with respect to the increasing distance of sensors from the sink [36]. Moreover, a good idea for further work should be that of comparing the performance of the protocols proposed in the present paper with that devised in [34]. Indeed, the training protocol of Section 3 presents irregular toggling between sleep and wake periods, which depends on the protocol computation, but in this way optimizes the sensor awake time. Its extension, shown in Section 4, presents sensor wake periods of fixed length and irregular sensor sleep periods. Hence, both versions consume energy in the synchronization between the sensors and the sink to handle irregular toggling between sleep and wake periods. In contrast, the asynchronous protocol proposed in [34] forces sensors to be awake and sleep for longer periods but avoids irregular toggling because sensors always alternate between awake and sleep periods both of fixed length.

Acknowledgment

The authors would like to thank Roger Wattenhofer for providing Reference [4].

Appendix

Proof of Lemma 3.1. 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. 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,

$$p(v) = 1 + \sum_{j=1}^{i-1} c_j. \tag{7}$$

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

$$p(u) = p(v) + \begin{cases} 1 & \text{if } u \text{ is the left child of } v \\ \frac{h}{2^{i-1}} & \text{otherwise.} \end{cases} \tag{8}$$

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 (7) and (8) 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.

Proof of Lemma 3.2. The proof is by induction on the depth i of node u in T'. To settle the basis, observe 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. 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, $d_1, d_2, \ldots, d_{i-1}$. By the inductive hypothesis,

$$n(v) = h + \sum_{j=1}^{i-1} d_j. \tag{9}$$

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

$$n(u) = n(v) + \begin{cases} -\frac{h}{2^i} & \text{if } u \text{ is the left child of } v \\ +\frac{h}{2^i} & \text{otherwise.} \end{cases} \tag{10}$$

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 (9) and (10) 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.

Proof of Lemma 4.1. The proof is by induction on the depth i of node u in T'. To settle the basis, let $i = 1$ and assume u to be the j-th child of the root. Then, $B(u) = j$ and $b_1, \ldots, b_{\log q}$ give by definition the binary representation of $j - 1$. Hence, $B(u) = \sum_{h=1}^{\log q} b_h 2^{\log q - h} + 1$. For the inductive step, assume the statement true for all nodes in T' of depth less than the depth of u. Let v be the parent of u. By inductive assumption

$$B(v) = \sum_{d=1}^{i-1} 2^{(i-1-d)\log q} \left(\sum_{h=1}^{\log q} b_{(d-1)\log q + h} 2^{\log q - h} + 1 \right).$$

By Eq. (3), $B(u) = qB(v) + j$ if u is the j-th child of v, while by definition $b_{(i-1)\log q + 1}, \ldots, b_{i \log q}$ give the binary representation of $j - 1$. Therefore,

$$\begin{aligned} B(u) &= q \left(\sum_{d=1}^{i-1} 2^{(i-1-d)\log q} \left(\sum_{h=1}^{\log q} b_{(d-1)\log q + h} 2^{\log q - h} + 1 \right) \right) + j \\ &= 2^{\log q} \left(\sum_{d=1}^{i-1} 2^{(i-1-d)\log q} \left(\sum_{h=1}^{\log q} b_{(d-1)\log q + h} 2^{\log q - h} + 1 \right) \right) + \sum_{h=1}^{\log q} b_{(i-1)\log q + h} 2^{\log q - h} + 1 \\ &= \sum_{d=1}^{i} 2^{(i-d)\log q} \left(\sum_{h=1}^{\log q} b_{(d-1)\log q + h} 2^{\log q - h} + 1 \right). \end{aligned}$$

Hence the lemma is proved.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramanian, E. Cayirci, Wireless sensor networks: A survey, Computer Networks 38 (4) (2002) 393–422.
[2] S. Bandyopadhyay, E. Coyle, An efficient hierarchical clustering algorithm for wireless sensor networks, in: Proc. IEEE INFOCOM 2003, San Francisco, CA, April 2003.
[3] A.A. Bertossi, S. Olariu, M.C. Pinotti, Efficient training of sensor networks, in: Proc. ALGOSENSORS 2006, in: LNCS, vol. 4240, 2006, pp. 1–12.
[4] N. Burri, P. von Rickenbach, R. Wattenhofer, Dozer: Ultra-low power data gathering in sensor networks, in: Proc. IPSN'07, Cambridge, MA, April 2007.
[5] D. Culler, D. Estrin, M. Srivastava, Overview of sensor networks, IEEE Computer 37 (8) (2004) 41–49.
[6] K.A. Delin, S.P. Jackson, The sensor web: A new instrument concept, in: Proc. SPIE Symposium on Integrated Optics, San Jose, CA, January 2001.
[7] D.M. Doolin, N. Sitar, Wireless sensors for wild remonitoring, in: Proc. SPIE Symposium on Smart Structures and Materials, (NDE 2005), San Diego, CA, March 6–10, 2005.
[8] D. Estrin, D. Culler, K. Pister, G. Sukhatme, Instrumenting the physical world with pervasive networks, Pervasive Computing 1 (1) (2002) 59–69.
[9] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next century challenges: Scalable coordination in sensor networks, in: Proc. ACM MOBICOM, Seattle, WA, August 1999.
[10] S. Ghiasi, A. Srivastava, X. Yang, M. Sarrafzadeh, Optimal energy-aware clustering in sensor networks, Sensors 2 (2002) 258–269.
[11] B. Hemingway, W. Brunette, T. Anderal, G. Boriello, The flock: Mote sensors sing in undergraduate curriculum, IEEE Computer 37 (8) (2004) 72–78.

[12] Intel Research, New computing frontiers — the Wireless vineyard, http://www.intel.com/technology/techresearch/research/rs01031.htm accessed December 19 2006.
[13] K. Langendoen, N. Reijers, Distributed localization algorithm, in: R. Zurawski (Ed.), Embedded Systems Handbook, CRC Press, Boca Raton, FL, 2004.
[14] K. Martinez, J.K. Hart, R. Ong, Sensor network applications, IEEE Computer 37 (8) (2004) 50–56.
[15] D. Nicolescu, B. Nath, Ad-hoc positioning system, in: IEEE GlobeCom, November 2001.
[16] S. Olariu, A. Waada, L. Wilson, M. Eltoweissy, Wireless sensor networks leveraging the virtual infrastructure, IEEE Network 18 (4) (2004) 51–56.
[17] S. Olariu, I. Stojmenovic, Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting, in: Proc. IEEE INFOCOM, Barcelona, April 2006.
[18] J. Polastre, R. Szewcyk, A. Mainwaring, D. Culler, J. Anderson, Analysis of wireless sensor networks for habitat monitoring, in: C.S. Raghavendra, K.M. Sivalingam, T. Znati (Eds.), Wireless Sensor Networks, Kluwer Academic, 2004.
[19] S. Roundy, P.K. Wright, J. Rabaey, Energy Scavenging for Wireless Sensor Networks with Special Focus on Vibrations, Kluwer Academic Press, 2004.
[20] K. Ryokai, J. Cassell, StoryMat: A play space for collaborative storytelling, in: Proc. CHI'99, October 1999.
[21] C. Savarese, K. Langendoen, J. Rabaey, Robust positioning algorithms for distributed ad hoc wireless sensor networks, in: USENIX Technical Annual Conference, Monterey, CA, June 2002.
[22] A. Savvides, L. Girod, M. Srivastava, D. Estrin, Localization in sensor network, in: C.S. Raghavendra, K.M. Sivalingam, T. Znati (Eds.), Wireless Sensor Networks, Kluwer Academic, 2004.
[23] A. Savvides, H. Park, M. Srivastava, The bits and flops of the n-hop multilateration primitive for node localization problems, in: First ACM Int'l Workshop on Wireless Sensor Networks and Application, WSNA, Atlanta, September 2002.
[24] R.C. Shah, J. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in: Proc. IEEE WCNC, Orlando, FL, March 2002.
[25] N.S. Shenck, J.A. Paradiso, Energy scavenging with shoe-mounter piezoelectrics, IEEE Micro 21 (2001) 30–41.
[26] V. Shnayder, M. Hempstead, B. Chen, G.W. Allen, M. Welsh, Simulating the power consumption of large-scale sensor network applications, in: Proc. SenSys'04, Baltimore, MD, November 2004.
[27] M. Sichitiu, C. Veerarithiphan, Simple accurate synchronization for wireless sensor networks, in: Proc. WCNC'03, 2003.
[28] F. Sivrukaya, B. Yener, Time synchronization in sensor networks: A survey, IEEE Network 18 (4) (2004) 45–50.
[29] K. Sohrabi, J. Gao, V. Ailawadhi, G. Pottie, Protocols for self-organization of a wireless sensor network, IEEE Personal Communications 7 (5) (2000) 16–27.
[30] M. Srivastava, R. Muntz, M. Potkonjak, Smart kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments, in: Proc. ACM MOBICOM, Rome, Italy, July 2001.
[31] R. Szewczyk, J. Polastre, A. Mainwaring, J. Anderson, D. Culler, An analysis of a large scale habitat monitoring application, in: Proc. 2nd ACM Conference on Embedded Networked Sensor Systems, November 2004.
[32] R. Szewczyk, E. Osterweil, J. Polatre, M. Hamilton, A. Mainwaring, D. Estrin, Habitat monitoring with sensor networks, Communications of the ACM 47 (6) (2004) 34–40.
[33] A. Waada, S. Olariu, L. Wilson, M. Eltoweissy, K. Jones, Training a wireless sensor network, Mobile Networks and Applications 10 (1) (2005) 151–168.
[34] Q. Xu, R. Ishak, S. Olariu, S. Salleh, On asynchronous training in sensor networks, in: 3rd Intl. Conf. on Advances in Mobile Multimedia, K.Lumpur, September 2005.
[35] V.V. Zhirnov, D.J.C. Herr, New frontiers: Self-assembly and nano-electronics, IEEE Computer 34 (1) (2001) 34–43.
[36] G. Zhou, T. He, S. Krishnamurthy, J.A. Stankovic, Models and solutions for radio irregularity in wireless sensor networks, ACM Transactions on Sensor Networks 2 (2) (2006) 221–262.