

Asynchronous Training in Wireless Sensor Networks

F. Barsi* A.A. Bertossi† F. Betti Sorbelli* R. Ciotti* S. Olariu‡
M.C. Pinotti*

Abstract

Scalable energy-efficient training protocols are proposed for massively-deployed sensor networks, where sensors are initially anonymous and unaware of their location. The protocols are based on an intuitive coordinate system imposed onto the deployment area which partitions the anonymous sensors into clusters. The protocols are asynchronous, in the sense that the sensors wake up for the first time at random, then alternate between sleep and awake periods both of fixed length, and no explicit synchronization is performed between them and the sink. Theoretical properties are stated under which the training of all the sensors is possible. Moreover, a worst-case analysis as well as an experimental evaluation of the performance is presented, showing that the protocols are lightweight and flexible.

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, 24]. Together with innovative and focused network design techniques that will make possible massive deployment [22] 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 [14, 17], to smart learning environments [4, 9, 16], to habitat monitoring [15, 19], to environment monitoring [12, 20], to greenhouse and vineyard experiments [5], to forest fire detection [6], to helping the elderly and the disabled [12, 18], among others.

The peculiar characteristics of sensor networks pose unique challenges to the design of protocols. First of all, the limited energy 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, for reasons of scalability, that no sensor knows the topology of the network.

*Department of Computer Science and Mathematics, University of Perugia, 06123 Perugia, Italy, {barsi,pinotti}@unipg.it

†Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy, bertossi@cs.unibo.it

‡Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA, olariu@cs.odu.edu

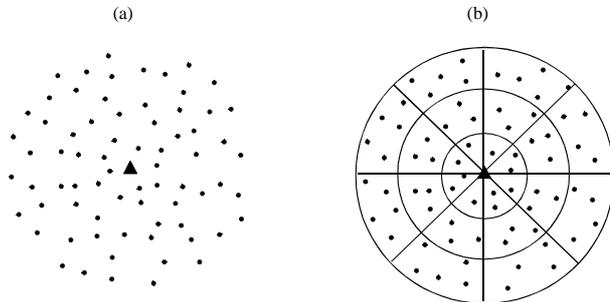


Figure 1: (a) *A sensor network with a central sink.* (b) *The trained sensor network.*

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 sinks, 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 ρ , 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).

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 [10, 13]. The latter task, referred to as *training*, has been considered in several recent papers by Olariu *et al.* [3, 14, 21, 23]. In particular, they devised some training protocols for sensor networks, which differ on whether or not sensors need some kind of explicit synchronization with the sink. Such training protocols have different performance, measured in terms of total time for training, overall sensor awake time, and number of sensor sleep/wake transitions. In particular, the model in [23] assumes that the sink and the sensors are asynchronous, in the sense that the sensors wake up for the first time at random and then alternate between sleep and awake periods both of fixed length, while no explicit synchronization is performed between them and the sink.

The main contribution of this paper is to further study the task of training, assuming the same asynchronous model as that originally defined in [23]. The present paper completes the work of [23], by stating novel theoretical properties under which the training of all the sensors in the network is possible. Moreover, new protocols are presented which are lightweight in terms of both the number of sleep/wake transitions and the overall sensor awake time for training.

The remainder of this paper is organized as follows. Section 2 discusses the wireless sensor network model 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

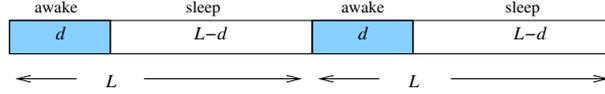


Figure 2: *The sensor sleep-awake cycle.*

sink, as first suggested in [21]. Section 3 is the backbone of the entire paper, presenting the theoretical underpinnings of a basic training protocol, called Flat-, along with its worst-case performance analysis. Section 4 shows two variants of the basic protocol, called Flat and Flat+, as well as a two-level approach, which improve the Flat- performance. Section 5 presents an experimental evaluation of the performance, tested on randomly generated instances, confirming the analytical results in the worst case and showing a much better behaviour in the average case. Finally, Section 6 offers concluding remarks.

2 The network model

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

A sensor is a device that possesses three basic capabilities: sensory, computation, and wireless communication, and operates 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 and a transmission range of r ;
- c. In order to save energy, each sensor alternates between *sleep* periods and *awake* periods, as depicted in Figure 2 – the sensor sleep-awake cycle is of total length L out of which the sensor is in sleep mode for $L - d$ time and in awake mode for d time;
- d. Each sensor is asynchronous – it wakes up for the first time according to its internal clock and is not engaging in an explicit synchronization protocol with either the sink or the other sensors;
- e. Each sensor has no global information about the network topology, but can hear transmissions from the sink;
- f. Individual sensors must work *unattended* – once deployed it is either infeasible or impractical to devote attention to individual sensors.

The task of 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, 7, 17]. 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 involves establishing [21]:

1. *Coronas*: The deployment area is covered by k coronas C_0, C_1, \dots, C_{k-1} determined by k concentric circles, centered at the sink, whose radii are $0 < r_0 < r_1 < \dots < r_{k-1} = \rho$;
2. *Wedges*: The deployment area is ruled into a number of equiangular wedges, centered at the sink, which are established by directional transmission [14].

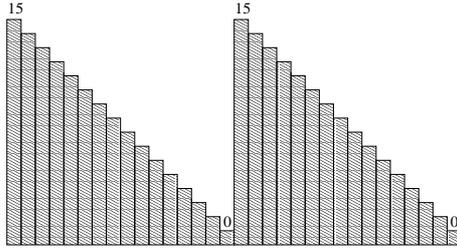


Figure 3: *The sink transmission cycle.*

For the sake of simplicity, in this paper, it is assumed that the corona width is equal to the sensor transmission range r , and hence the (outer) radius r_i of corona C_i is equal to $(i + 1)r$. Moreover, the length L of the sensor sleep-awake cycle is assumed to be no smaller than the number k of coronas. 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. In particular, a cluster is the locus of all nodes having the same coordinates in the coordinate systems [14].

3 The Flat– protocol

The main goal of this section is to present the details of the basic corona training protocol (the wedge training protocol is similar and will not be discussed), where each individual sensor has to learn the identity of the corona to which it belongs, regardless of the moment when it wakes up for the first time. To see how this is done, it is useful to assume the time ruled into slots. The sensors and the sink use equally long, in phase slots, but they do not necessarily start counting the time from the same slot.

The idea of the protocol is illustrated in Figure 3. Immediately after deployment the sink cyclically repeats a transmission cycle which involves k broadcasts at successively lower power levels. Each broadcast lasts for a slot and transmits a beacon equal to the identity of the outmost corona reached. Precisely, the sink starts out by transmitting the beacon $k - 1$ at the highest power, sufficient to reach the sensors up to the outmost corona C_{k-1} ; next, the sink transmits the beacon $k - 2$ at a power level that can be received up to corona C_{k-2} , but not by the sensors in corona C_{k-1} . For the subsequent $k - 2$ slots, the sink continues to transmit at decreasing power levels until it concludes its transmission cycle with a broadcast that can be received only by the sensors in corona C_0 . In general, at time slot τ , with $\tau \geq 0$, the sink transmits the beacon $k - 1 - |\tau|_k$ with a power level that can reach all the sensors up to corona $C_{k-1-|\tau|_k}$, where $|a|_b$ stands for the non negative remainder of the integer division between a and b (i.e. $|a|_b$ is the same as a modulo b). The sink transmission cycle is repeated for a time τ_1 sufficient to accomplish the entire corona training protocol. The protocol for the sink is shown in Figure 4.

In order to describe the protocol for sensors, it is crucial to point out that each sensor is aware of the sink behaviour and of the total number k of coronas. Immediately after deployment, each sensor wakes up at random within the 0-th and the $(k - 1)$ -th time slot and starts listening to the sink for d time slots (that is, its awake period). Then, the sensor goes back to sleep for $L - d$ time slots (that is, its sleep period). Such a sleep/wake transition will be repeated until

| |
|--|
| <pre> Procedure Sink (k, τ_1); for $\tau := 0$ to $\tau_1 - 1$ do transmit the beacon $k - 1 - \tau _k$ up to corona $C_{ k-1-\tau _k}$; </pre> |
|--|

Figure 4: *The protocol for the sink.*

the sensor will learn the identity of the corona to which belongs, that is, until the sensor will be trained. Each sensor, during the training process, uses a k -bit register R to keep track of the beacons, i.e. corona identities, transmitted by the sink while the sensor is awake. As soon as the sensor hears a sink transmission for the first time, it starts to fill the register R and it is able to learn the sink global time t within the current sink transmission cycle, that is $t = \lceil \tau \rceil_k$. From now on, such a time will regularly increase so that the sensor can derive from t the beacon $|k - 1 - t|_k$ that the sink is transmitting. Then, in each time slot when the sensor is awake, one entry of R can be always set either to 0 or to 1. In fact, if the sensor hears beacon c , then it sets $R_c = 1$, while if the sensor hears nothing, it sets $R_{|k-1-t|_k} = 0$. Note that the awake sensors which belong to corona c , with $c > 0$, are able to receive any transmitted beacon from c up to $k - 1$, whereas they cannot hear the beacons from 0 up to $c - 1$. Hence, if a sensor sets $R_c = 0$ (resp., $R_c = 1$) then it belongs to a corona whose identity is higher than (resp., smaller than or equal to) c . Note that only the sensors in corona 0 can hear beacon 0 and thus they are the only ones which can set $R_0 = 1$. From the above discussion, the following *training condition* holds:

Lemma 3.1. [23] *A sensor which belongs to corona c , with $c > 0$, is trained as soon as the entries R_c and R_{c-1} of its register R are set to 1 and 0, respectively. A sensor which is in corona 0 is trained as soon as R_0 is set to 1.*

The resulting sensor protocol, called Flat-, is illustrated in Figure 5. Procedure Flat- mimics the behaviour of the sensor from its first wakeup until it is trained, that is when the identity of the corona to which it belongs is stored into *mycorona*. Each sensor counts the number ν of sleep/wake transitions needed to be trained (line 1), it keeps its local time t , which is initialized when the sensor receives a beacon for the first time from the sink (that is, when *heard* is set to **true** in line 7), and it stores in *alarm-clock* the time when the next sleep/wake transition is planned (line 21–23). After any entry of R is filled, the sensor checks the training condition stated in Lemma 3.1. Observe that lines 12–19 cannot be executed when $c = k - 1$, because the beacon $k - 1$ reaches the outmost corona C_{k-1} , all awake sensors hear, and thus they execute lines 6–11. In the procedure, each sensor executes $O(1)$ arithmetic/logic operations per time slot.

In the following, some conditions on the parameters k, L , and d will be investigated which guarantee that all the sensors are trained, independent of their first wakeup time and from the corona c they belong to. Hereafter, let (a, b) denote the *greatest common divisor* between a and b . Moreover, if $(a, b) = 1$, let $\frac{1}{a}|_b$ be the multiplicative inverse of a modulo b (e.g. see [8]).

Lemma 3.2. *Fixed L, d , and k , there are exactly $k' = \frac{k}{(L, k)}$ different corona identities that can be transmitted by the sink when the sensor starts any awake period. Assuming that the sensor wakes up for the first time at slot x , $0 \leq x \leq k - 1$, then the corona identity transmitted when the sensor starts its i -th awake period is $|K_x - i(L, k)|_{k'}|_k = |K_x - |i|_{k'}(L, k)|_{k'}|_k$, where K_x is the corona identity transmitted at time x , that is $K_x = C_{|k-1-x|_k}$. Such k' coronas identities can be reindexed as $|K_x - s(L, k)|_k$, for $0 \leq s \leq k' - 1$.*

```

Procedure Flat- ( $k, L, d$ );
1   heard := trained := false;  $\nu := 0$ ;
2   while wakeup and  $\neg$  trained do
3      $\nu := \nu + 1$ ;
4     for  $i := 0$  to  $d - 1$  do
5       if received beacon  $c$  then
6         if  $\neg$  heard then
7           heard := true,  $t := k - 1 - c$ ;
8            $R_c := 1$ ;
9           if  $c = 0$  or ( $R_c = 1$  and  $R_{c-1} = 0$ ) then
10            mycorona :=  $c$ , trained := true;
11             $t := t + 1$ ;
12          else
13            if heard then
14               $c := k - 1 - |t|_k$ ;
15               $R_c := 0$ ;
16              if  $R_{c+1} = 1$  then
17                mycorona :=  $c$ , trained := true;
18                 $t := t + 1$ ;
19            if heard then
20              alarm-clock :=  $t := t + L - d$ ;
21            else
22              alarm-clock := alarm-clock +  $L$ ;
23            go to sleep until the alarm-clock rings;
24

```

Figure 5: The Flat- protocol for a sensor.

Proof. Consider a sensor that wakes up for the first time at the global time slot $\tau = x$, while the sink is transmitting the beacon $K_x = |k - 1 - \tau|_k = |k - 1 - x|_k$. The i -th sleep-awake cycle of such a sensor starts at time $x + iL$ while the sink is transmitting the beacon $|k - 1 - x - iL|_k = |K_x - i|L|_k|_k$, with $i \geq 0$. Observe that L and k can be rewritten as $L = gL'$ and $k = gk'$, where $g = (L, k)$. Since $|L|_k = |gL'|_k = g|L'|_{k'}$, one has $|K_x - i|L|_k|_k = |K_x - ig|L'|_{k'}$. The same corona is transmitted again at the beginning of any two awake periods of the sensor which are k' apart because $|K_x - (i + k')|L|_k|_k = |K_x - (i + k')g|L'|_{k'}$ and $|K_x - (ig + k'g)|L'|_{k'}$ are distinct and differ by a multiple of g . Hence, overall there are exactly k' different coronas which can be transmitted by the sink when the sensor starts its awake periods, independent of how long the training process will be. In addition, such k' corona identities can be rearranged so that in the new order two consecutive coronas differ exactly by g . Indeed the s -th corona in the new order, that is $|K_x - sg|_k$, corresponds to the first beacon transmitted in the $\lfloor \frac{s}{L'} \rfloor_{k'}$ -th awake period, with $0 \leq s \leq k' - 1$. \square

Therefore, after exactly k' sleep-awake cycles, that is after $k'L$ time slots, which correspond to $\frac{k'L}{k} = \frac{k'L}{gk'} = \frac{L}{g} = L'$ sink transmission cycles, the behaviour of the sensor and the sink will be cyclically repeated. In other words, at the beginning of the k' -th awake period, the sensor and the sink are in the same reciprocal state as they were at the beginning of the 0-th one, with the only difference that, if the sensor can be trained, it has heard the sink at least once. Thus:

Lemma 3.3. *Fixed L, d , and k , all the entries of R that the sensor can fill are set within the first $\frac{2k}{(L,k)}$ sleep-awake cycles.*

Proof. During the first $k' = \frac{k}{(L,k)}$ awake periods of any sensor, the sink transmits no more than $\frac{k}{(L,k)}d$ different corona identities. These corona identities will be cyclically transmitted during the training process of such a sensor. They correspond to exactly all the positions of R that the sensor can set and they include all the beacons that the sensor can hear from the sink. Hence, in the worst case, the sensor needs $k' = \frac{k}{(L,k)}$ awake periods to hear the sink for the first time and further $k' = \frac{k}{(L,k)}$ awake periods to fill R . \square

Clearly, if the training condition of Lemma 3.1 cannot be verified by a sensor within its first $2k' = \frac{2k}{(L,k)}$ sleep-awake cycles, such a sensor will never be trained, independent of how long the training process will continue. The following result shows under which conditions for k , L and d all the sensors can be trained and also gives an upper bound on the number of sleep-awake cycles needed to accomplish the entire training process.

Theorem 3.4. *All the sensors are trained in at most $2k' = 2\frac{k}{(L,k)}$ sleep-awake cycles if and only if $d \geq (L, k)$.*

Proof. For brevity let $g = (L, k)$. By contradiction, suppose that all the sensors have been trained and let $d < g$. By Lemmas 3.2 and 3.3, in at most $\frac{2k}{g}$ sleep-awake periods, each sensor has filled at most $k'd$ entries of R . Since $d < g$, each sensor has filled less than k entries of R . Such filled entries depend on the time slot x when the sensor woke up for the first time. Consider now all the sensors that woke up at the same time x . Note that they have filled, although with different configurations, the same positions of R independent of the corona they belong. Let c be one unfilled entry of R . By the hypothesis of massive random deployment, there is at least one sensor that woke up at time x in each corona, and hence at least one sensor in corona c . Clearly, such a sensor will not be trained because the training condition in Lemma 3.1 will be never satisfied.

Conversely, if $d \geq g$, by Lemma 3.2, in k' consecutive sleep-awake cycles, the beacons transmitted by the sink in the first slot of such k' cycles are exactly g apart. Since an awake period lasts $d \geq g$ slots, at least g new corona identities are transmitted by the sink during an awake period of the sensor. Hence, after having heard the sink within the first k' awake periods, the sensor fills at least g entries of R in each awake period and completely fills R in at most other k' awake periods. Therefore, the sensor is trained in at most $2k'$ consecutive awake periods. Note that this happens for all the sensors, independent of their first wake-up time x and of the corona c to which they belong. \square

In the following, a better bound on the maximum number of sleep-awake periods required in the worst case to train a sensor is discussed for two particular cases, namely, $d = (L, k)$ and $d = |L|_k$. Note that, since $d = |L|_k = (L, k)|L'|_{k'}$, Theorem 3.4 holds in both cases and hence register R is completely filled within the first $\frac{2k}{(L,k)}$ sleep-awake periods. However, the training condition may be verified earlier because it is sufficient that the entries c and $c - 1$ of R be filled. Precisely, Lemmas 3.5 and 3.6 specify, for $d = (L, k)$ and $d = |L|_k$ respectively, in which awake period of a sensor that wakes up for the first time at slot x the sink is transmitting an arbitrary beacon c .

Lemma 3.5. *Let c be any corona identity and assume $d = (L, k)$. The sink transmits the beacon c during the $i_{c,x}$ -th awake period of a sensor that wakes up for the first time at slot x , where $i_{c,x} = \left\lceil \left\lfloor \frac{|K_x - c|_k}{d} \right\rfloor \left\lceil \frac{1}{L'} \right\rceil \right\rceil_{k'}$, $L' = \frac{L}{d}$, and $k' = \frac{k}{d}$.*

Proof. When the sensor wakes up at time x the sink is transmitting the beacon K_x . Moreover, the beacon values decrease within a sink transmission cycle. Thus, the beacon c will be transmitted, starting from K_x , during the j -th group of d consecutive corona identities such that $j = \lfloor \frac{|K_x - c|_k}{d} \rfloor$. Such a j -th group of d consecutive corona identities will be transmitted during the $i_{c,x}$ -th sensor awake period in which the sink transmits $\left| K_x - \lfloor \frac{|K_x - c|_k}{d} \rfloor d \right|_k$ as the first beacon. Hence, by Lemma 3.2, $i_{c,x}$ is derived by solving the equation $|K_x - i_{c,x}(L, k)|L'|_{k'}|_k = \left| K_x - \lfloor \frac{|K_x - c|_k}{d} \rfloor d \right|_k$. Since $d = (L, k)$, it follows $i_{c,x} = \left\lfloor \frac{|K_x - c|_k}{L'} \right\rfloor_{k'}$. \square

Lemma 3.6. *Let c be any corona identity and assume $d = |L|_k$. The sink transmits the beacon c during the $i_{c,x}$ -th awake period of a sensor which wakes up for the first time at slot x , where $i_{c,x} = \left\lfloor \frac{|K_x - c|_k}{d} \right\rfloor$.*

Proof. The proof is similar to that of Lemma 3.5. Only observe that now, since $d = |L|_k = (L, k)|L'|_{k'}$, $i_{c,x}$ is derived by solving the equation $|i_{c,x}(L, k)|L'|_{k'}|_k = |jd|_k$, and hence $i_{c,x} = j$. \square

The following two lemmas determine when the training condition is satisfied by a sensor.

Lemma 3.7. *Let $d = (L, k)$. A sensor which wakes up for the first time at slot x and belongs to corona c , with $c > 0$, is trained during the i -th awake period where $i = i_{c-1,x}$, if $i_{c,x} \leq i_{c-1,x}$, or $i \leq i_{c,x} + \lfloor \frac{1}{L'} \rfloor_{k'}$, if $i_{c,x} > i_{c-1,x}$. If $c = 0$, then $i = i_{0,x}$.*

Proof. If $i_{c,x} \leq i_{c-1,x}$, during the $i_{c,x}$ awake period the sensor hears the beacon c and hence it sets $R_c = 1$. Moreover, during the $i_{c-1,x}$ awake period, the sensor sets $R_{c-1} = 0$ because it does not hear $c - 1$ but, having already heard c , it knows what the sink is transmitting. If $i_{c,x} > i_{c-1,x}$, in the worst case the sensor hears for the first time during the $i_{c,x}$ -th awake period and sets $R_c = 1$. Then, the beacon $c - 1$ will be transmitted at the i -th awake period such that $|K_x - i(L, k)|L'|_{k'}|_k = |K_x - (j + 1)d|_k$, where $j = \left\lfloor \frac{|K_x - c|_k}{d} \right\rfloor$. Solving the above equation, one has $i = |(j + 1) \lfloor \frac{1}{L'} \rfloor_{k'}|_{k'} = i_{c,x} + \lfloor \frac{1}{L'} \rfloor_{k'}$. \square

Lemma 3.8. *Let $d = |L|_k$. A sensor which wakes up for the first time at slot x and belongs to corona c , with $c > 0$, is trained during the i -th awake period where $i = i_{c-1,x}$, if $i_{c,x} \leq i_{c-1,x}$, or $i \leq i_{c,x} + 1$, if $i_{c,x} > i_{c-1,x}$. If $c = 0$, then $i = i_{0,x}$.*

Proof. The proof is similar to that of Lemma 3.7. For $d = |L|_k$, only observe that, when $i_{c,x} > i_{c-1,x}$, since $d = |L|_k = (L, k)|L'|_{k'}$, i is derived by solving the equation $|i(L, k)|L'|_{k'}|_k = |(j + 1)d|_k$, and hence $i = j + 1$. \square

In order to analytically evaluate the performance of the Flat- training protocol, let us introduce the following notations. Let ν be the number of sleep/wake transitions required by a sensor to be trained in the worst case, that is the number of sleep/wake transitions required to fill the whole register R . Moreover, let ω be the overall sensor awake time and τ be the total time for training. Recalling that a sleep-awake period has length L , a sensor is awake for d time slots per sleep-awake period, and wakes up at time $x < k$, one has $\omega = \nu d$ and $\tau = \nu L + k$. Thus, the worst case performance for the Flat- protocol can be summarized as follows:

Theorem 3.9. *Fixed L , d , and k , if $d < (L, k)$ then there are sensors which cannot be trained by the Flat- protocol; otherwise all the sensors are trained, and:*

1. If $(L, k) \leq d < |L|_k$, then $\nu \leq \frac{k}{(L, k)} + \left\lfloor \frac{1}{L'} \right\rfloor_{k'}$, where $k' = \frac{k}{(L, k)}$ and $L' = \frac{L}{(L, k)}$;
2. If $|L|_k \leq d < k$, then $\nu \leq \left\lfloor \frac{k}{|L|_k} \right\rfloor + 1$;
3. If $d = k$, then $\nu \leq 2$.

Proof. When $(L, k) \leq d < |L|_k$, since by Lemma 3.2 the k' coronas transmitted by the sink when the sensor wakes up do not depend on d , the sensor cannot be trained later than in the case $d = (L, k)$, because the register R is filled faster. Hence by Lemma 4.3, observing that $\left\lfloor \frac{|K_x - c|_k}{d} \right\rfloor$ varies between 0 and $\frac{k}{d} - 1$, one has $\nu \leq \frac{k}{(L, k)} + \left\lfloor \frac{1}{L'} \right\rfloor_{k'}$, where $k' = \frac{k}{(L, k)}$ and $L' = \frac{L}{(L, k)}$. Similarly, when $|L|_k \leq d < k$, the sensor cannot be trained later than in the case $d = |L|_k$. Hence, by Lemma 3.8, $\nu \leq \left\lfloor \frac{k}{|L|_k} \right\rfloor + 1$. Note that, when k is a multiple of $|L|_k$, $\nu = \left\lfloor \frac{k}{|L|_k} \right\rfloor + 1$ only for those sensors that wake up for the first time while the sink is transmitting corona $c - 1$ and they belong to corona c . Finally, when $d = k$, two sleep-awake cycles are needed only by those sensors which wake up for the first time while the sink is transmitting corona $c - 1$ and which belong to corona c . \square

Note that, when $d = (L, k)$ or $d = |L|_k$, since ν equals the upper bound stated in Theorem 3.9, $\tau = \left(\frac{k}{(L, k)} + \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \right) L + k$ or $\tau = \left(\left\lfloor \frac{k}{|L|_k} \right\rfloor + 1 \right) L + k$, respectively. Referring to Figure 4, it should be clear that τ_1 must be an upper bound on the total time for training, which is derived as $\tau_1 = \nu L + k$ by choosing ν according to the upper bounds given by Theorem 4.4.

4 Improvements

The Flat- protocol presented in the previous section can be improved in several ways so as to reduce the number ν of sleep/wake transitions, and hence also the overall sensor awake time as well as the total time for training.

4.1 The Flat protocol

As a first improvement of Flat-, recall that, as soon as a sensor hears the sink transmission for the first time, it learns from the beacon the sink global time modulo the sink transmission cycle. Therefore, it can immediately retrieve backwards the coronas which it did not hear and which were transmitted by the sink during its previous awake periods, setting to 0 the corresponding entries of R . The resulting improved protocol, which is called *Flat*, is derived from the Flat- protocol by modifying, as shown in Figure 6, the `if` instruction in lines 6-7 of Figure 5. As a drawback, a sensor may now execute as many as $O(\nu d)$ arithmetic/logic operations per time slot. The time required to perform such arithmetic operations, however, should be negligible with respect to the time slot length, which instead depends on the characteristics of the radio broadcast equipment.

Observed that, when the sensor hears the sink for the first time, it fills R as it would have heard the sink since the first time it woke up, Lemma 3.3 and Theorem 3.4 can be restated as follows:

Lemma 4.1. *Fixed L, d , and k , all the entries of R the sensor can fill are set within the first $k' = \frac{k}{(L, k)}$ sleep-awake cycles.*

| | |
|-----|--|
| 6 | if \neg heard then |
| 7 | heard := true, $t := k - 1 - c$; |
| 7.1 | for $j := i - 1$ downto 0 do $R_{ c+1+j _k} := 0$; |
| 7.2 | for $j := \nu - 1$ downto 1 do |
| 7.3 | for $h := d - 1$ downto 0 do $R_{ c+j _k+i-h _k} := 0$; |

Figure 6: The extra instructions for the Flat protocol.

Theorem 4.2. All the sensors are trained in at most $k' = \frac{k}{(L,k)}$ sleep-awake cycles if and only if $d \geq (L, k)$.

In other words, the Flat protocol completes the training process in at most k' sleep/wake transitions. Such a bound is tight in the particular case that $d = (L, k)$, while it can be lowered when $d = |L|_k$. Indeed, since Lemmas 3.5 and 3.6 still hold, Lemmas 3.7 and 3.8 can be restated for the Flat protocol as follows:

Lemma 4.3. When $d = (L, k)$ or $d = |L|_k$, a sensor which wakes up for the first time at slot x and belongs to corona c is trained during the i -th awake period where $i = \max\{i_{c-1,x}, i_{c,x}\}$, if $c > 0$, or $i = i_{0,x}$, if $c = 0$.

Proof. When $i_{c,x} \leq i_{c-1,x}$, the proof is the same as that in Lemmas 3.7 and 3.8 for $d = (L, k)$ and $d = |L|_k$, respectively. When $i_{c,x} > i_{c-1,x}$, although in the worst case the sensor hears for the first time during the $i_{c,x}$ -th awake period, since R is set backwards, both $R_c = 1$ and $R_{c-1} = 0$ are set during such awake period. \square

The worst case performance for the Flat protocol is summarized below:

Theorem 4.4. Fixed L , d , and k , if $d < (L, k)$ then there are sensors which cannot be trained by the Flat protocol; otherwise all the sensors are trained, and:

1. If $(L, k) \leq d < |L|_k$, then $\nu \leq \frac{k}{(L,k)}$;
2. If $|L|_k \leq d < k$, then $\nu \leq \left\lceil \frac{k}{|L|_k} \right\rceil$;
3. If $d = k$, then $\nu = 1$.

Proof. When $d = (L, k)$, by Lemmas 3.5 and 4.3, a sensor is trained in at most $\frac{k}{(L,k)}$ sleep/wake transitions. Similarly, when $d = |L|_k$, the result derives from Lemmas 3.6 and 4.3. Moreover, when $(L, k) < d < |L|_k$ and $|L|_k < d < k$, the sensor cannot be trained later than in the case $d = (L, k)$ and $d = |L|_k$, respectively, because by Lemma 3.2 the k' coronas transmitted by the sink when the sensor wakes up are the same (such coronas depend only on L and k) and clearly the register R is filled faster. Finally, when $d = k$ at least one sleep/wake transition is needed. Note that, in both cases $d = k$ and k multiple of $|L|_k$, the sensors that belong to corona c and wake-up when the sink is transmitting corona $c - 1$ will set R_{c-1} the first time they hear the sink, without waiting that the sink retransmits beacon $c - 1$, saving one transition with respect to Flat-. \square

Note that, when $d = (L, k)$ and $d = |L|_k$, $\tau = \frac{kL}{(L,k)} + k$ and $\tau = \left\lceil \frac{k}{|L|_k} \right\rceil L + k$, respectively, because ν matches the upper bound given in Theorem 4.4.

4.2 The Flat+ protocol

A further improvement to the Flat protocol exploits the fact that when a sensor hears a beacon c , it knows that it will also hear all the beacons greater than c , and thus it can immediately set to 1 the entries from R_c up to R_{k-1} . Similarly, when a sensor sets an entry R_c to 0, it knows that it cannot hear any beacon smaller than c , and thus it can immediately set to 0 the entries from R_{c-1} down to R_0 , too. In contrast to the previous protocols, the sensor now fills entries of R relative to beacons not yet transmitted during its awake periods. Therefore, it can look ahead to decide whether it is worthy or not to wake up in the next awake period. If the d entries of R that will be transmitted by the sink in the next awake period have already been filled, then the sensor can skip its next awake period, thus saving energy. The sensor repeats the look ahead process above until at least one unfilled entry is detected among the d entries corresponding to a future awake period. The resulting protocol, called Flat+, is illustrated in Figure 7. Procedure Flat+ makes use of two variables, $\max0$ and $\min1$, which record the largest (smallest, resp.) index of R which has been filled to 0 (1, resp.). When a beacon c is heard, the sensor sets to 1 all the entries from R_c to $R_{\min1}$ (line 11). When an entry R_c has to be set to 0, then all the entries from $R_{\max0}$ to R_c are set to 0 (line 19). When the sensor hears the sink for the first time, it stores in $\max0$ the largest entry of R which must be 0 due to its previous awake periods (lines 8-9), and thus it sets to 0 the entries from R_0 to $R_{\max0}$ (line 10). Finally, at the end of the awake period, the sensor performs the above mentioned look ahead process, properly setting the alarm-clock (lines 24–29).

Clearly, the number ν of sleep/wake transitions of Flat+ cannot be larger than that of Flat. Moreover, when $d = |L|_k$ or $d = (L, k)$, one can find bad instances where ν , in the worst case, is the same for both Flat+ and Flat. For example, when $d = |L|_k$, a sensor which uses Flat+, belongs to corona c , and wakes up when the sink transmits $K_x = c - 1$ can never take advantage from the look ahead processing. Hence, the results in Theorem 4.4 hold for both the Flat and Flat+ protocols. However, as it will be experimentally checked in Section 5, the average behaviour of Flat+ is much better than that of Flat, specially for small values of d .

4.3 The two-level approach

The protocols discussed so far can be further improved by following a nesting approach in which the k coronas are viewed as k_1 macrocoronas of k_2 adjacent coronas each. Precisely, each sensor first learns in which macrocorona it belongs and then refines its training by determining the microcorona inside its macrocorona. Once a sensor learns the index of its macrocorona, say m with $0 \leq m \leq k_1 - 1$, as well as that of its microcorona, say μ with $0 \leq \mu \leq k_2 - 1$, it obtains its actual corona identity as $c = k_2m + \mu$, where of $0 \leq c \leq k_1k_2 - 1$. For determining both the macrocorona and microcorona identities, any of the Flat protocol variants can be used.

The protocol for the sink is shown in Figure 8. The sink works in two levels and counts the total time in τ . In the first level, the sink cyclically repeats the *macrocorona transmission cycle*, that is a cycle of length k_1 using decreasing powers so as to distinguish different consecutive macrocoronas. In fact, at time slot $z = 0$, the sink starts out by transmitting the beacon $k_1 - 1$ to a power sufficient to reach the sensors up to the outmost macrocorona, that is up to corona C_{k-1} . Next, the sink transmits the beacon $k_1 - 2$ at a power that can be received up to the $(k_1 - 2)$ -th macrocorona, that is corona C_{k-k_2-1} . For the subsequent $k_1 - 2$ slots, the sink continues to transmit at decreasing powers until it concludes its cycle at time slot $z = k_1 - 1$

```

Procedure Flat+ ( $k, L, d$ );
1   heard := trained := false;  $\nu := 0$ ; max0 := 0; min1 :=  $k - 1$ ;
2   while wakeup and  $\neg$  trained do
3      $\nu := \nu + 1$ ;
4     for  $i := 0$  to  $d - 1$  do
5       if received beacon  $c$  then
6         if  $\neg$  heard then
7           heard := true,  $t := k - 1 - c$ ;
8           max0 :=  $\max\{\max0, |c + i|_k\}$ ;
9           for  $j := \nu - 1$  downto 0 do max0 :=  $\max\{\max0, |c + j|_k + i|_k\}$ ;
10          for  $h := 0$  to max0 do  $R_h := 0$ ;
11          for  $h := c$  to min1 do  $R_h := 1$ ;
12          min1 :=  $c$ ;
13          if  $c = 0$  or ( $R_c = 1$  and  $R_{c-1} = 0$ ) then
14            mycorona :=  $c$ , trained := true;
15             $t := t + 1$ ;
16          else
17            if heard then
18               $c := k - 1 - |t|_k$ ;
19              for  $h := \max0$  to  $c$  do  $R_h := 0$ ;
20              max0 :=  $c$ ;
21              if  $R_{c+1} = 1$  then
22                mycorona :=  $c$ , trained := true;
23                 $t := t + 1$ ;
24            if heard then
25              filled:=true, alarm-clock :=  $t := t + L - d$ ;
26              while filled do
27                 $s := |k - 1 - t|_k$ ,  $z := 0$ ;
28                while  $z \leq d - 1$  and  $R_{|s-z|_k}$  has been filled do  $z := z + 1$ ;
29                if  $R_z$  is unfilled then filled := false else alarm-clock :=  $t := t + L$ ;
30            else
31              alarm-clock := alarm-clock +  $L$ ;
32            go to sleep until the alarm-clock rings;

```

Figure 7: *The Flat+ protocol for a sensor.*

```

Procedure Sink ( $k_1, k_2, \tau_1, \tau_2$ );
 $\tau := 0$ ;
for  $z := 0$  to  $\tau_1 - 1$  do
  transmit the beacon  $|k_1 - 1 - z|_{k_1}$  up to corona  $C_{k_2(|k_1-1-z|_{k_1}+1)-1}$ ;
   $\tau := \tau + 1$ ;
for  $m := 0$  to  $k_1 - 1$  do
  for  $z := 0$  to  $\tau_2 - 1$  do
    transmit the beacon  $|k_2 - 1 - z|_{k_2}$  up to corona  $C_{mk_2+|k_2-1-z|_{k_2}}$ ;
     $\tau := \tau + 1$ ;

```

Figure 8: *The two-level protocol for the sink.*

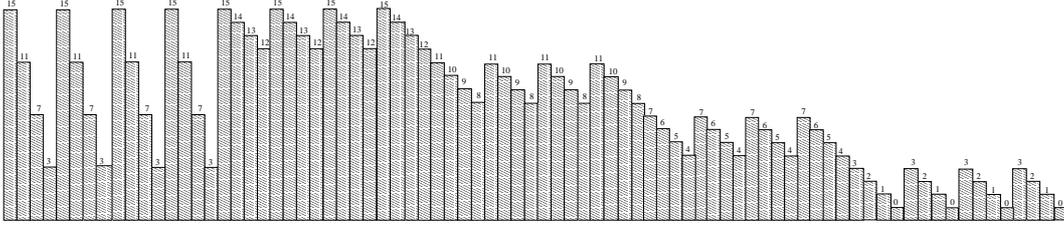


Figure 9: *The sink transmission cycle for the TwoLevel and TwoLevel+ protocols.*

with a broadcast that can be received only by the sensors in the 0-th macrocorona, that is, up to corona C_{k_2-1} . The first level lasts for τ_1 time slots, thus repeating $\frac{\tau_1}{k_1}$ times the macrocorona transmission cycle. The time τ_1 is properly chosen to allow all the sensors to be trained with respect to their macrocorona.

In the second level, for each macrocorona, the sink cyclically repeats a *microcorona transmission cycle*, that is one of length k_2 using decreasing powers so as to distinguish different consecutive coronas. Such a microcorona transmission cycle is repeated $\frac{\tau_2}{k_2}$ times so as to allow all the sensors in each macrocorona m to be also trained with respect to their microcorona. Overall the second level of the protocol lasts $k_1\tau_2$ time slots.

As regard to the protocol for the sensors, it is assumed that each sensor is aware of the two-level sink behaviour and thus of the numbers k_1 and k_2 of macrocoronas and microcoronas, respectively. Each sensor wakes up at time x , with $0 \leq x \leq \min\{k_1, k_2\}$, and repeats its sleep-awake cycle of length L such that $L \geq d \geq \max\{(L, k_1), (L, k_2)\}$. Each sensor uses a k_1 -bit register P and a k_2 -bit register Q to keep track of the macrocorona and microcorona identities, respectively. As soon as the sensor wakes up at time x , it performs one of the protocol variants, i.e. Flat-, Flat, and Flat+, using its register P to learn its macrocorona identity m . When it has been trained on its macrocorona, it sets its alarm clock to $\tau_1 + (k_1 - 1 - m)\tau_2 + x$ to be ready for the training on its microcorona, and goes to sleep. Reawakened, the sensor performs again the same protocol variant, but now filling its register Q to learn its microcorona identity μ . Clearly, as soon as it knows both m and μ , it derives its corona identity $c = k_2m + \mu$, and thus it is trained.

Depending on which protocol, Flat-, Flat, and Flat+, is used to train the sensors on each macrocorona and microcorona level, three two-level protocols are achieved, denoted by TwoLevel-, TwoLevel, and TwoLevel+. In Figure 9, the macrocorona and microcorona sink transmission cycles are depicted for the TwoLevel and TwoLevel+ protocols in the case where $k = 16$, $k_1 = k_2 = 4$, $L = 6$, $d = |L|_4 = 2$. In this case, $\nu_1 = \nu_2 = \frac{4}{2}$, and hence $\tau_1 = 2 * 6 + 4$, $\tau_2 = 2 * 6 + 4$, and $\tau = 16 + 4 * 16 = 80$. Note that the sink performs $\frac{\tau_1}{k_1} = 4$ sink transmission cycles of length k_1 to train the sensors on their macrocorona, and additional $\frac{\tau_2}{k_2} = 4$ sink transmission cycles of length k_2 per macrocorona to train them on their microcorona.

In general, with respect to the performance of the two-level protocols, one has:

Theorem 4.5. *Fixed L , d , k , k_1 , and k_2 , with $k = k_1k_2$ and $L \geq d \geq \max\{(L, k_1), (L, k_2)\}$, letting ν_1 and ν_2 be, respectively, the numbers of sensor sleep/wake transitions required to train a sensor on k_1 macrocoronas and k_2 microcoronas, the two-level protocols require $\nu = \nu_1 + \nu_2$ sleep/wake transitions and $\omega = (\nu_1 + \nu_2)d$ overall sensor awake time. Moreover, the total time for training is $\tau = \tau_1 + \tau_2k_1$, where τ_1 and τ_2 must be the upper bounds on the total time required*

by the training protocol adopted on each level.

Note that, by Theorems 3.9 and 4.4, tight bounds on the values of τ_1 and τ_2 can be derived only when $d = (L, k)$ and $d = |L|_k$. In all other cases, the total time of each level is derived from $\tau = \nu L + k$, setting ν equal to the upper bound given in Theorems 3.9 and 4.4. For example, consider the TwoLevel- protocol and assume $d = |L|_{k_1}$, and $(L, k_2) \leq d < |L|_{k_2}$. Then, $\tau_1 = \left(\left\lfloor \frac{k_1}{|L|_{k_1}} \right\rfloor + 1\right)L + k_1$ and $\tau_2 = \left(\frac{k_2}{(L, k_2)} + \left\lfloor \frac{1}{L'} \right\rfloor_{k'}\right)L + k_2$, where $k' = \frac{k_2}{(L, k_2)}$ and $L' = \frac{L}{(L, k_2)}$.

Next, the worst case performance of the Flat protocol is compared with that of the corresponding TwoLevel protocol when the same value of L and d are used, $\frac{k_1}{(L, k_1)} \neq 1$, $\frac{k_2}{(L, k_2)} \neq 1$, and $(L, k_2) \neq 1$. Note that to satisfy the constraints of both the Flat and TwoLevel protocols, d must vary between $(L, k) = (L, k_1)(L, k_2)$ and $\min\{k_1, k_2\}$, and L must be greater than k . When $d = (L, k)$, by Lemma 4.1 and Theorem 4.4, the number of transitions is at most $\frac{k_1}{(L, k_1)} + \frac{k_2}{(L, k_2)}$ for TwoLevel and is at least $\frac{k}{(L, k)}$ for Flat. Since $(L, k) = (L, k_1)(L, k_2)$, one has $\frac{k_1}{(L, k_1)} + \frac{k_2}{(L, k_2)} < \frac{k_1}{(L, k_1)} \frac{k_2}{(L, k_2)} = \frac{k}{(L, k_1)(L, k_2)} = \frac{k}{(L, k)}$. Similarly, TwoLevel beats Flat when $d = \min\{k_1, k_2\}$. Indeed letting $d = k_1 = \min\{k_1, k_2\}$, TwoLevel requires at most $1 + \frac{k_2}{(L, k_2)} < k_2$ sleep/wake transitions, while Flat needs at least $\frac{k}{d} = k_2$ transitions. Since in both protocols the number of transitions decreases when d increases, TwoLevel beats Flat when $(L, k) \leq d \leq \min\{k_1, k_2\}$. Finally it is easy to see that both the overall sensor awake time and the total time for training of Flat are larger than those of TwoLevel.

5 Experimental tests

In this section, the worst and average performance of the corona training protocols are experimentally tested. The algorithms were written in C++ and the experiments were run on an AMD Athlon X2 4800+ with 2 GB RAM. In the simulation, each corona has a unit width. There are $N = 10000$ sensors uniformly distributed within a circle, centered at the sink, having radius $\rho = k$. Precisely, the polar coordinates of each sensor are generated choosing at random two real numbers. The first one, uniformly distributed between 0 and k , represents the radial coordinate of the sensor, that is, its distance from the sink. The second number, uniformly distributed between 0 and 2π , represents the angular coordinate of the sensor, that is, the positive angle required to reach the sensor from the polar axis. In the experiments, both the worst and average number of transitions, denoted by ν_{\max} and ν_{avg} , as well as both the worst and average overall sensor awake time, ω_{\max} and ω_{avg} , are evaluated. Such average values are obtained by summing up the values for each single sensor and then dividing by the number of sensors. Moreover, the total time τ , which measures the time required to terminate the whole training process, is evaluated.

Consider first the experiments for the Flat-, Flat, and Flat+ protocols. In the simulations, the number k of coronas is fixed to 64. The length L of the sensor sleep-awake cycle assumes the values 104 and 168. Finally, the sensor awake period d is an integer that varies, with a step of 4, between the greatest common divisor $(L, k) = 8$ and $k = 64$, thus including $|L|_k = 40$. The results are reported only when all the sensors can be trained, that is for $d \geq 8$, and are averaged over 3 independent experiments.

Figure 10 shows the number ν_{\max} and ν_{avg} of transitions for the different values of d . According to Theorems 3.9 and 4.4, when $d = 8$, Flat- has $\nu_{\max} = \frac{k}{(L, k)} + \left\lfloor \frac{1}{L'} \right\rfloor_{k'} = 8 + 5 = 13$,

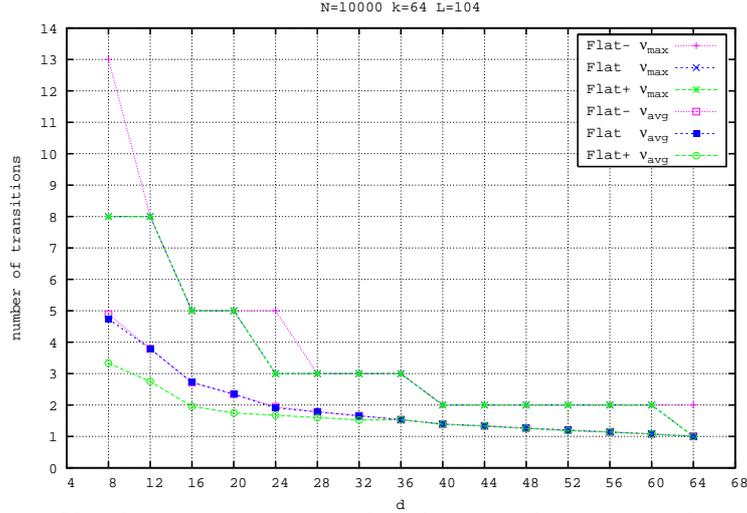


Figure 10: Number of transitions when $k = 64$, $L = 104$, and $8 \leq d \leq 64$.

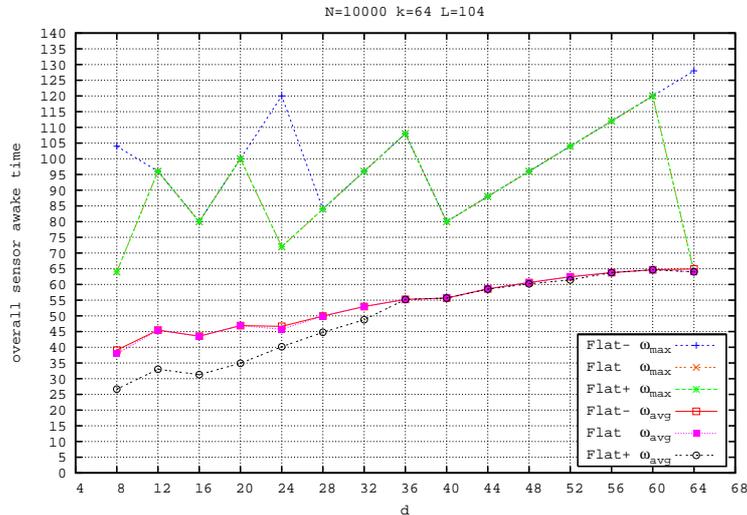


Figure 11: Overall sensor awake time when $k = 64$, $L = 104$, and $8 \leq d \leq 64$.

while both Flat and Flat+ have $\nu_{\max} = \left\lceil \frac{k}{|L|_k} \right\rceil = 8$. Similarly, when $d = 40$, all protocols take $\nu_{\max} = 2$ transitions. Except for the extreme values $d = 8$ and $d = 64$, the greatest percentage of gain for ν_{\max} is achieved when $d = 24$, where both Flat+ and Flat employ forty percent less transitions than Flat-. As regard to the average performance, one notes that ν_{avg} is considerable better than ν_{\max} for all three protocols. Flat and Flat- have almost the same average performances, while Flat+ always behaves better than them. In particular, its greatest percentage of gain for ν_{avg} is obtained in the range $8 \leq d \leq 20$, where Flat+ improves about twenty/thirty percent upon Flat-.

Figure 11 shows the awake times $\omega_{\max} = \nu_{\max}d$ and $\omega_{\text{avg}} = \nu_{\text{avg}}d$, which measure the overall energy spent by each sensor to be trained. Although the number of transitions decreases as d increases, Figure 11 suggests to choose a small value of d from the sensor awake time perspective. The minimum ω_{\max} is achieved by Flat and Flat+ for $d = 8$ and $d = 64$, as expected by Theorems 3.9 and 4.4. However, when $d = 8$, ω_{avg} lowers to about two thirds of ω_{\max} for Flat- and Flat, and to about one third for Flat+. Note that Flat+ has the maximum gain when d is small. Indeed, it can fill the same entries of R just listening to the sink for

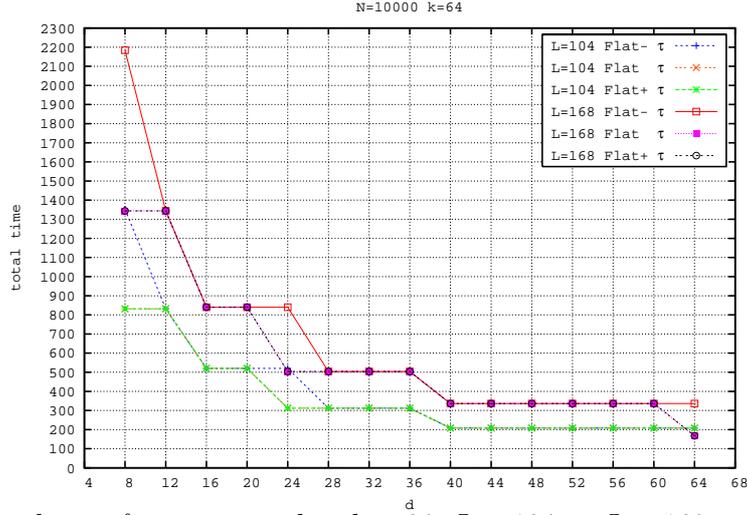


Figure 12: Total time for training when $k = 64$, $L = 104$ or $L = 168$, and $8 \leq d \leq 64$.

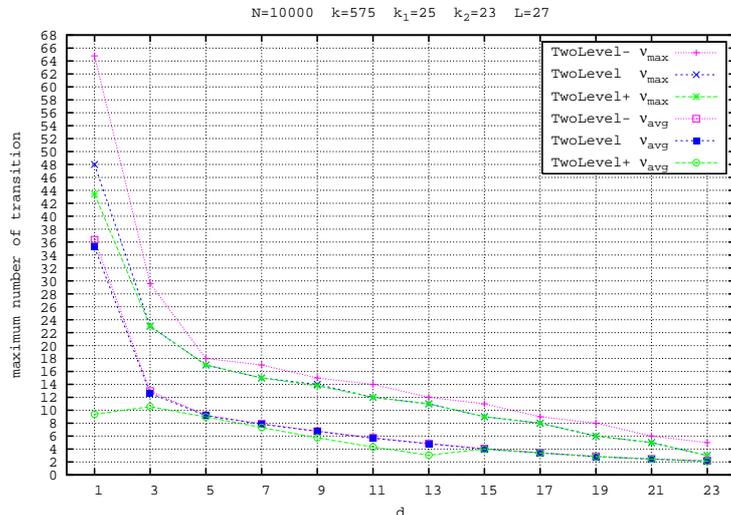


Figure 13: Number of transitions when $k = 575$, $k_1 = 25$, $k_2 = 23$, $L = 27$, and $1 \leq d \leq 23$.

a single slot or for d slots. Hence, small values of d save the same number of transitions as larger values, but allow sensors to reduce their energy consumption because they stay awake for smaller periods.

Figure 12 exhibits the total time τ required to accomplish the entire training task, for both $L = 104$ and $L = 168$. Since $|168|_{64} = |104|_{64} = 40$, by Lemma 3.2, each protocol maintains the same behaviour with respect to the number of transitions. Thus, the plots for $L = 168$ of ν_{\max} and ν_{avg} , and hence of ω_{\max} and ω_{avg} , are exactly the same as those shown in Figures 10 and 11. Recalling that $\tau = \nu_{\max}L + k$, the total time for $L = 168$ scales by a constant $\sim \frac{168}{104}$, as depicted in Figure 12. In general, all values of L such that $|L|_k$ is the same present the properties above, namely, ν and ω are identical, while τ scales. Therefore, the minimum total time τ is achieved for the smallest value of L . However, larger values of L could be also selected in order to increase the longevity of the wireless sensor network. Fixed d , a longer L results in a longer life as the life of a sensor is measured in terms of the overall number of sleep-awake cycles until its energy is exhausted.

Consider now the experiments relative to the two-level approach. Recall that TwoLevel-

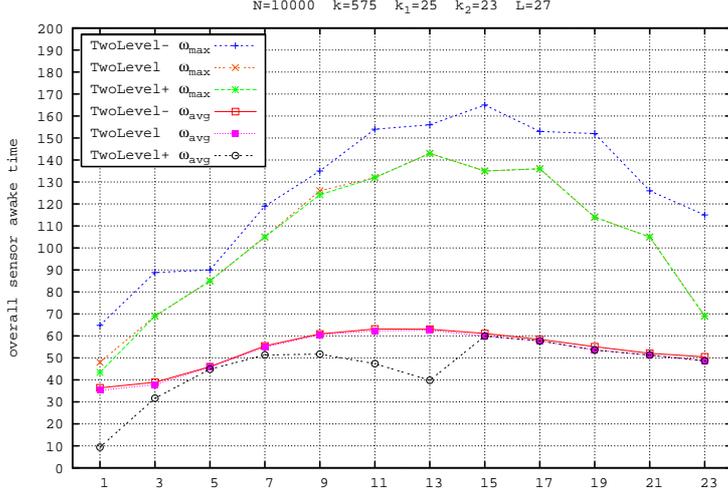


Figure 14: Overall sensor awake time when $k = 575$, $k_1 = 25$, $k_2 = 23$, $L = 27$, and $1 \leq d \leq 23$.

TwoLevel, and TwoLevel+ denote, respectively, the protocol when Flat-, Flat, and Flat+ are employed on each single level. In the simulations, the number k of coronas is fixed to 575 while k_1 and k_2 are fixed to 25 and 23, respectively. The length L of the sensor sleep-awake cycle is fixed to 27 and the sensor awake period d varies, with a step of 2, between $\max\{(L, k_1), (L, k_2)\} = 1$ and $\min\{k_1, k_2\} = 23$. The results are averaged over 3 independent experiments.

Figures 13, 14, and 15 plot both the average and worst case performance of ν , ω , and τ . As explained in the previous section, one can easily derive the worst case performance of the two-level protocols in Figure 13 from the worst case performance of the one-level protocols. For example, when $d = (L, k_1) = (L, k_2) = 1$, TwoLevel- requires $\nu = 65$ sleep/wake transitions because the Flat- protocol requires $\nu_1 = \frac{k_1}{(L, k_1)} + \left\lfloor \frac{1}{L} \right\rfloor_{k_1} = 25 + 15 = 38$ transitions when $k_1 = 25$ and $\nu_2 = \frac{k_2}{(L, k)} + \left\lfloor \frac{1}{L} \right\rfloor_{k_2} = 23 + 6 = 19$ transitions when $k_2 = 23$.

Figure 14 shows the awake times $\omega_{\max} = \nu_{\max}d$ and $\omega_{\text{avg}} = \nu_{\text{avg}}d$. The curves in Figure 14 smoothly change, without the abrupt peaks of Figure 11, because now the number of transitions monotonically decreases, as shown in Figure 13. Moreover, note that when $d = (L, k) = 1$ the overall awake time is minimum, although the number of transitions is maximum, and TwoLevel+ reaches the maximum gain with respect to the other algorithms, in both the worst and average cases. For $d = 13$, the TwoLevel+ protocol registers on both ν and ω a local minimum.

Figures 16, 17, and 18 are devoted to compare the behaviour of the Flat and TwoLevel protocols. As before, $k = 575$, $k_1 = 25$, and $k_2 = 23$. The length L of the sensor sleep-awake cycle is fixed to 27 and 577 for TwoLevel and Flat, respectively. Note that both such values of L are the smallest possible choices for the two protocols, because Flat requires L larger than k and TwoLevel needs L larger than $\max\{k_1, k_2\}$. The sensor awake period d varies, with a step of 5, between $(L, k) = 1$ and 76. The results are averaged over 3 independent experiments.

As expected for $1 = (L, k) \leq d \leq 23 = \min\{k_1, k_2\} = 23$, the TwoLevel protocol always significantly beats the Flat protocol. Note that, in contrast to Flat, TwoLevel cannot be employed when $d \geq 23$. Observe that $\nu = O(1)$ can be achieved by both the Flat and TwoLevel protocols in correspondence of $d = \Theta(k)$ and $d = \Theta(\sqrt{k})$, respectively, leading therefore to a big difference in the values of ω and τ .

Although two different values of L are used in the experiments, $L = 577$ could be also used for the TwoLevel protocol. In such a case, since $\lfloor 577 \rfloor_{25} = \lfloor 27 \rfloor_{25}$, the number of transitions

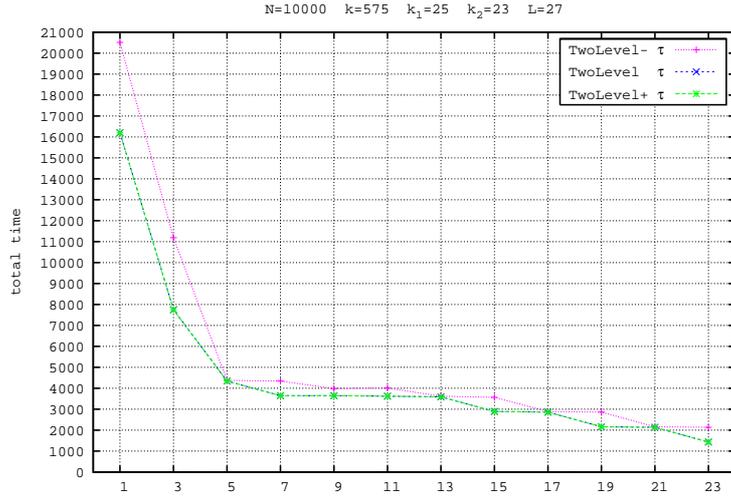


Figure 15: Total time for training when $k = 575$, $k_1 = 25$, $k_2 = 23$, $L = 27$, and $1 \leq d \leq 23$.

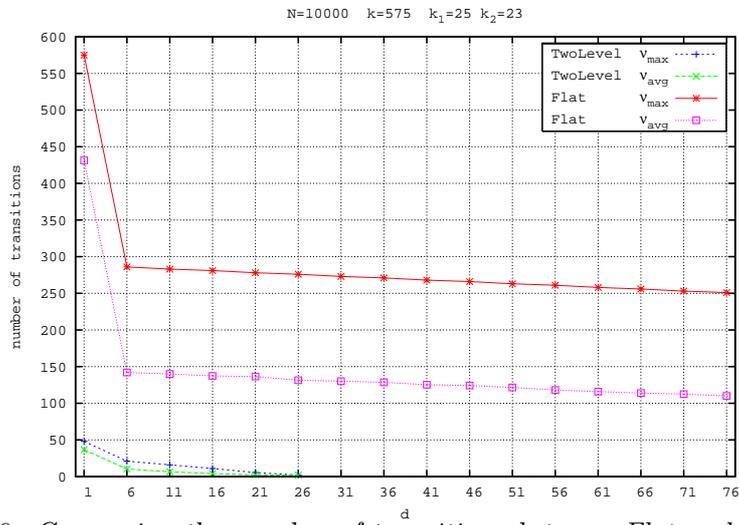


Figure 16: Comparing the number of transitions between Flat and TwoLevel.

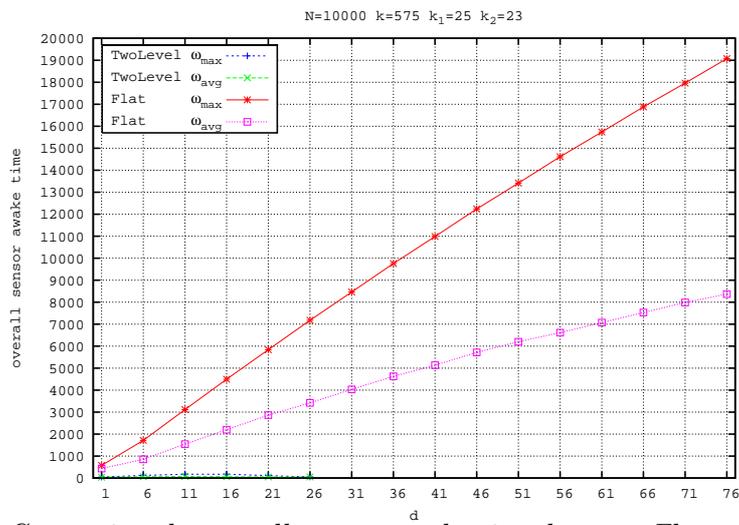


Figure 17: Comparing the overall sensor awake time between Flat and TwoLevel.

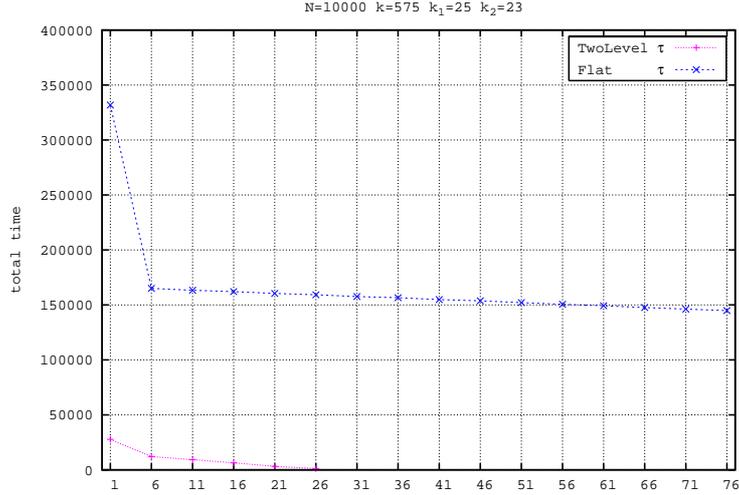


Figure 18: Comparing the total time for training between Flat and TwoLevel.

ν_1 on the macrocoronas would remain the same, whereas the number of transitions ν_2 on the microcoronas would become slightly larger because $|577|_{23} = 2 < |27|_{23} = 4$. Therefore, $\nu = \nu_1 + \nu_2$ would increase at most by $\lceil \frac{23}{2} \rceil - \lceil \frac{23}{4} \rceil = 6$ with respect to that shown in Figure 16. Obviously the total time τ for TwoLevel with $L = 577$ would dramatically increase with respect to that illustrated in Figure 18, but it would still remain below that of Flat.

6 Concluding remarks

In this work new protocols have been proposed which employ the asynchronous model originally presented in [23] and are lightweight in terms of the number of sleep/wake transitions and overall sensor awake time for training. Among the various protocol variants and improvements, Flat+ is the simplest one from the computational viewpoint because each sensor performs $O(1)$ operations per time slot. In contrast, TwoLevel+ has the best performance, but all the two-level protocols as well as Flat+ cannot be used if the sensor is not allowed to skip one or more awake periods.

The results presented in this paper show that the protocols are flexible, in the sense that their parameters can be properly tuned. For instance, fixed the number k of coronas, one can decide the optimal values of d and L so as to minimize the number of sleep/wake transitions and/or the overall awake time per sensor. Conversely, one can fix the desired number of sleep/wake transitions, and then select suitable values of d and L .

However, several questions still remain open. In particular, a good idea for further work should be that of comparing the performance of the protocols proposed in the present paper with those devised in [3]. Indeed, the synchronous training protocols of [3] present an irregular toggling between sleep and wake periods, so as to optimize the total time for training, but they consume energy in the explicit synchronization between the sensors and the sink to handle such irregular sleep/wake toggling. In contrast, the asynchronous protocols proposed in the present paper assume sensors to periodically follow sleep-awake cycles, thus avoiding irregular toggling, but they take a larger total time for training.

References

- [1] I. 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. *Proc. IEEE INFOCOM 2003*, San Francisco, CA, April 2003.
- [3] A. A. Bertossi, S. Olariu, and M.C. Pinotti. Efficient training of sensor networks. *Proc. 2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks (AlgoSensors 2006)*. Lecture Notes in Computer Science 4240, 1–12, 2006.
- [4] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *IEEE Computer*, 37(8):41–49, 2004.
- [5] K. A. Delin and S. P. Jackson. The sensor web: a new instrument concept. *Proc. SPIE Symposium on Integrated Optics*, San Jose, California, January 2001.
- [6] D. M. Doolin and N. Sitar. Wireless sensors for wild remonitoring. *Proc. SPIE Symposium on Smart Structures and Materials (NDE 2005)*, San Diego, California, March 6-10, 2005.
- [7] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal energy-aware clustering in sensor networks. *Sensors*, 2:258–269, 2002.
- [8] H. Griffin, *Elementary Theory of Numbers*, McGraw Hill, New York, 1954.
- [9] B. Hemingway, W. Brunette, T. Anderal, and G. Boriello. The flock: Mote sensors sing in undergraduate curriculum. *IEEE Computer*, 37(8), 2004, 72-78.
- [10] K. Langendoen and N. Reijers. Distributed localization algorithm. In *Embedded Systems Handbook*, R. Zurawski (Editor), CRC Press, Boca Raton, FL, 2004.
- [11] J. J. Lee, B. Krishnamachari, and C. C. Jay Kuo. Impact of heterogeneous deployment on lifetime sensing coverage in sensor networks. *Proc. IEEE SECON*, 2004.
- [12] K. Martinez, J.K. Hart, and R. Ong. Sensor network applications. *IEEE Computer*, 37(8):50–56, 2004.
- [13] D. Nicolescu. Positioning in ad-hoc sensor networks. *IEEE Network*, 18(4):24–29, 2004.
- [14] S. Olariu, A. Waada, L. Wilson, and M. Eltoweissy. Wireless sensor networks leveraging the virtual infrastructure. *IEEE Network*, 18(4):51–56, 2004.
- [15] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson. Analysis of wireless sensor networks for habitat monitoring. In *Wireless Sensor Networks*, Raghavendra, Sivalingam, and Znati, Eds., Kluwer Academic, 2004, 399-423.
- [16] K. Ryokai and J. Cassell. StoryMat: A play space for collaborative storytelling. *Proc. CHI'99*, October 1999.
- [17] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5), 2000, 16–27.
- [18] M. Srivastava, R. Muntz, and M. Potkonjak. Smart Kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments. *Proc. ACM MOBICOM*, Rome, Italy, July 2001.
- [19] R. Szewczyk, J. Polastre, A. Mainwaring, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems*, Nov. 2004.
- [20] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6), (2004), 34–40.
- [21] 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.
- [22] B. Warneke, M. Last, B. Leibowitz, and K. Pister. SmartDust: communicating with a cubic-millimeter computer. *IEEE Computer*, 34(1):44–51, 2001.
- [23] Q. Xu, R. Ishak, S. Olariu, and S. Salleh. On asynchronous training in sensor networks. *Proc. 3rd Intl. Conf. on Advances in Mobile Multimedia*, K.Lumpur, September 2005.
- [24] V.V. Zhirnov and D.J.C. Herr. New frontiers: self-assembly and nano-electronics. *IEEE Computer*, 34(1):34–43, 2001.