

Asynchronous Corona Training Protocols in Wireless Sensor and Actor 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 wireless networks consisting of sensors and a single actor, where the 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 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 actor. Theoretical properties are stated under which the training of all the sensors is possible. Moreover, both a worst-case and an average case analysis of the performance, as well as an experimental evaluation, are presented showing that the protocols are lightweight and flexible.

Index Terms— Wireless sensor networks, actors, corona training, localization, network protocols, design and analysis of algorithms

I. 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 [2], [25]. 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 aggregating sensors into *wireless sensor networks*, which have a significant impact on a wide array of applications [7].

Recently, it has been recognized that it would be beneficial to augment massively deployed sensor networks by more powerful entities, equipped with better processing capabilities, higher transmission power, and longer battery life. This leads to a heterogeneous deployment including, alongside with the tiny sensors, some entities referred to as *actors* [1] or Aggregation and Forwarding Nodes (AFN) [14]. While the sensors are tasked mainly to sense their immediate neighbourhood, the actors behave as both *actuators* and *coordinators*. Indeed, they collect, aggregate and fuse the data harvested by the sensors, thus taking decisions in order to rapidly respond on the environment in a meaningful way. Moreover, actors may coordinate with each other and/or communicate with the outside world through a central entity called *sink* (which could be a base station, a patrol, or a satellite). The so augmented version of the sensor network results in a

F. Barsi, F. Betti Sorbelli, R. Ciotti, and M.C. Pinotti are with the Department of Computer Science and Mathematics, University of Perugia, 06123 Perugia, Italy, {barsi, pinotti}@unipg.it

A.A. Bertossi is with the Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy, bertossi@cs.unibo.it

S. Olariu is with the Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA, olariu@cs.odu.edu

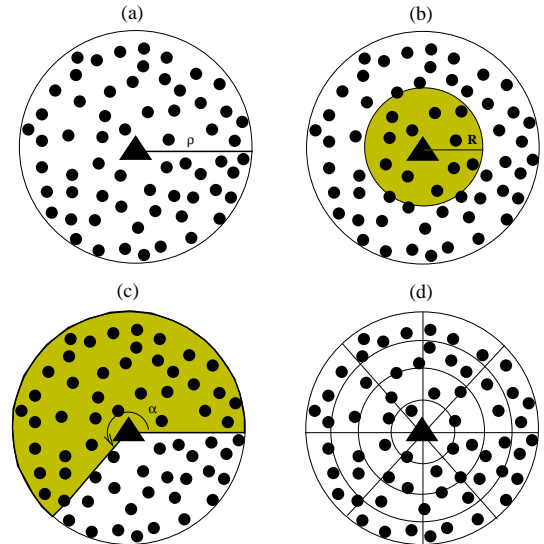


Fig. 1. An actor-centric subnetwork: (a) the sensors deployed in a circular field of radius ρ , (b) an actor broadcast of range R , (c) an actor broadcast of angle α , (d) the virtual coordinate system.

hierarchical network, commonly referred to as a wireless sensor and actor network (WSAN [1], or SANET [9]).

The typical mode of operation of an actor is to task the sensors in a circular field, centered at the actor itself, to produce data relevant to the mission at hand [14]. For instance, Figure I(a) illustrates the circular area of radius ρ monitored by an actor (depicted with a triangle), which defines an actor-centric subnetwork. In this scenario, the actor is equipped with a long-range radio with both isotropic and directional antennae. Specifically, by means of the isotropic antenna, the actor is able to send broadcasts with variable-range R to reach all the sensors at distance at most $R \leq \rho$ from the actor (see Fig. I(b)). Moreover, using the directional antenna, the actor can send full-range broadcasts (i.e., with $R = \rho$) to all the sensors lying in a circular sector of arbitrary angle α with respect to the polar axis (see Fig. I(c)). In general, a large sensor deployment area of any shape is organized into several cooperating actor-centric subnetworks, one for each deployed actor (where sensors lying in the intersection ranges of many actors should refer to just one actor, choosing one of them).

The random deployment results in sensors initially unaware of their spatial coordinates with respect to its referring actor. However, localization is a key factor for many wireless sensor network applications, like environmental monitoring, asset tracking, surveillance and disaster relief [12], [18]. In each case, the data gathered by the sensors are of scarce utility unless complemented with the location of the node that collected them. For example, knowing the sensory data position is critical for locating an

intruder vehicle or for guiding a team of first-aid on an emergency area. Moreover, understanding the sensory data may be location dependent, for instance, temperature and moisture values have different meanings depending on which area they refer to. The immediate approach to provide the exact geographic position of each sensor node is obviously based on the Global Positioning System (GPS). Although the GPS receivers are now common on cars, trucks, PDAs, and cell phones, they are unsuitable for massive deployed sensor networks due to the design constraints on the cost, dimension, and energy of each sensor node. Nonetheless, in such dense networks, most applications need to locate the small region where the event happens rather than the single positions of all the sensors that report the event itself. Hence, a coarse-grain location awareness [15], [21] is sufficient and the deployment area can be divided in small regions so that all the sensors deployed in one of such regions share the same virtual position and the data collected there can be used interchangeably or aggregated. Of course, there is a trade-off because the coarse-grain location awareness is lighterweight than GPS localization but the resulting positioning accuracy is only a rough approximation of the exact geographic location.

The task of allowing each sensor to acquire a coarse-grain location with respect to its referring actor is called *training* [23]. Since the aim of this paper is to study such a training task, it is assumed from now on that the whole network coincides with a single actor-centric subnetwork. Under this assumption, training consists in imposing a virtual coordinate system that divides the area into equiangular sectors and concentric coronas centered at the actor, as illustrated in Fig. I(d). After training, each sensor has learnt the corona and the sector to which it belongs. It is worth noting that, in this way, training provides for free a *clustering* of the sensors, where a cluster consists of all sensors having the same coordinates. Hence, on the top of training, all network protocols based on clustering can be used [2], [4], [8], [19]. For example, after training, routing can be easily performed as follows. Cluster-to-actor messages are trivially routed inward within a single sector, while cluster-to-cluster messages can be routed following several paths, e.g., first along the sector of the sender to reach the corona of the receiver, and then within such a corona (clockwise or counterclockwise, depending on which is the shortest path) to reach also the sector of the receiver [15]. In addition, to help the actor locating an event that happens in the network, each sensor can add its coordinates to the sensed data before delivering the messages to the actor.

While previous papers have studied the task of training in an actor-centric model where all the sensors are synchronized to the master clock running at the actor [5], [15], [21], the main contribution of this paper is to study training protocols in an asynchronous model, which only assumes very basic functionalities of the network – no explicit synchronization between the sensors and the actor is needed, and the sensors, which wake up for the first time at random, alternate between sleep and awake periods both of predefined lengths, established at the manufacturing time, and thus independent of the protocol computation. In fact, this is the behaviour of many actual sensors with harvesting capabilities which collect energy during the sleep periods and perform their duties during the awake periods [16], [24]. The present paper extends in a substantial way the work of [23] presenting three new *flat* protocols based on linear signal strength decrease. Moreover, three additional new protocols are exhibited which rely on a *two-*

level approach, where jumps are initially made in the flat protocols and are filled later on. Theoretical properties on the parameters of the training protocols are stated under which training of all the sensors in the network is possible. Moreover, a performance evaluation of the protocols is presented showing that they are lightweight in terms of both the number of wake/sleep transitions and the overall sensor awake time for training.

The remainder of this paper is organized as follows. Section II provides an overview of related works, Section III discusses the wireless sensor and actor network model and introduces the task of training, while Section IV offers a quick refresher of modular arithmetic. Section V is the backbone of the entire paper, presenting the theoretical underpinnings of a basic training protocol, called Flat–, along with its worst-case and average-case performance analysis. Section VI shows two variants of the basic protocol, called Flat and Flat+, as well as a two-level approach, which improve the Flat– performance. In particular, it is shown that the two-level protocols outperform the flat ones, lowering the number of transitions from a linear down to, at most, a square-root function. Section VII presents an experimental evaluation of the performance, tested on randomly generated instances, confirming the analytical results in both the worst and average cases, and showing a much better behaviour in practice. Finally, Section VIII offers concluding remarks.

II. RELATED WORK

In the literature, the task of determining an exact geographic location, referred to as *localization*, is recognized as a fundamental problem in designing sensor networks and has been extensively studied (see e.g. [3], [12], [18]). Since localization hardware, such as GPS [11], for each sensor is expensive in terms of cost and energy consumption, prominent solutions assume the existence of several anchor nodes which are aware of their location because they are the only nodes provided with GPS and allow other nodes to infer their locations by exchanging information with them. Localization algorithms can then be divided into two categories: *range-based* and *range-free* [10], [17]. In range-based algorithms, nodes estimate their distance to anchors, using some specialized hardware, and applying methods like triangulation or trilateration, which are based on the fact that a node position is uniquely determined when at least three anchor positions are available per node [3]. Other range-based algorithms use received radio signal strength, angle and/or time of arrival of signals, or difference of time of arrivals to locate the sensors. Although range-based algorithms result in a fine-grained localization scheme, all of them need some special hardware for the measurements at the sensors. On the other hand, range-free algorithms do not use any special hardware but accept a less accurate localization. For example, in the range-free *centroid* algorithm, the sensors receive the anchor positions, and using this proximity information, a simple centroid model is applied to estimate the position of the listening nodes [6]. Other solutions use methods similar to distance vector routing to allow the nodes to find the number of hops from the anchors. Anchors flood their location throughout the network maintaining a running hop-count at each node along the way. Nodes calculate their position based on the received anchor locations, on the hop-count from the corresponding anchor, and on the average-distance per hop [13]. In [10], an iterative method is pursued to narrow down the position accuracy until a tolerable error in the positioning is reached. In practice, each sensor repeatedly chooses

a triple of anchors from all audible anchors and tests whether it is inside the triangle formed by them, until all triples are exhausted or the required accuracy is achieved. At this point, the center of gravity of all of the triangles in which a node resides is assumed to be the sensor estimated position.

The localization algorithms discussed so far assume that the anchor nodes are special nodes mainly because they know their spatial coordinates. Instead, several recent papers [1], [5], [15], [21], [23] have considered the localization problem in a heterogeneous network whose anchor nodes, called actors, are provided with special transmission capabilities and steady power supply, while do not necessarily need GPS receivers. In such a context, localization is intended as the task of making each sensor able to acquire a coarse-grain location with respect to a given actor node and is referred to as training. The main characteristic of such training protocols relies on using a single actor node to impose a discretized polar coordinate system. The process is centralized and uses only asymmetric broadcasts (from the actor to the sensors) without multihop communications among the sensors. The sensors deduce their coarse-grain location exploiting the information received by the actor without performing any local communication. In particular, the two protocols presented in [5] assume that all the sensors are synchronized to the master clock running at the actor. Such two protocols can be thought as visits of complete trees, whose leaves represent coronas/sectors, whose node preorder/BFS numbers are related to the time slots, and whose node inorder/BFS numbers are related to the actor transmission ranges/angles, respectively. Exploiting the fully synchronized model and the capability of irregularly alternating between sleep and awake periods, whose frequency and length depend on the protocol computation, such protocols achieve an optimal time (in the number of coronas/sectors) for terminating the training process.

III. THE NETWORK MODEL

In this work, a wireless sensor and actor network is assumed that consists of a single, fixed actor, centrally placed with respect to a set of sensors randomly deployed in a circular field within the actor, as illustrated in Figure I(a).

It is assumed that the time is ruled into slots. The sensors and the actor use equally long, in-phase slots, but they do not necessarily start counting time from the same slot.

A sensor is a device that possesses three basic capabilities: sensory, computation, and wireless communication, and operates subject to the following fundamental constraints:

- a. Each sensor alternates between *sleep* periods and *awake* periods – the sensor sleep-awake cycle has a total length of L time slots, out of which the sensor is in sleep mode for $L - d$ slots and in awake mode for d slots;
- b. 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 actor or the other sensors;
- c. Individual sensors work *unattended* – once deployed it is either infeasible or impractical to devote attention to individual sensors;
- d. No sensor has global information about the network topology, but can hear transmissions from the actor;
- e. The sensors are *anonymous* – to assume the simplest sensor model, sensors do not need individually unique IDs;

- f. Each sensor has a modest non-renewable energy budget and a limited transmission range.

As shown in Figure I(d), training imposes a virtual coordinate system onto the sensor network by establishing:

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 actor, whose radii are $0 < r_0 < r_1 < \dots < r_{k-1} = \rho$;
2. *Sectors*: The deployment area is ruled into h equiangular sectors S_0, S_1, \dots, S_{h-1} , centered at the actor, each having a width of $\frac{2\pi}{h}$ radians.

For the sake of simplicity, in this paper, it is assumed that all the coronas and all the sectors have the same width, although this is not strictly required. In a practical setting, the corona width might be equal to the sensor transmission range, say r , and hence the (outer) radius r_i of corona C_i might be equal to $(i + 1)r$. In such a case, then, the corona number plus one gives the number of hops needed for a sensor-to-actor communication. Moreover, a sector S_j might consist of the portion of the deployment area between the two directional transmission angles $j\frac{2\pi}{h}$ and $(j + 1)\frac{2\pi}{h}$. 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 sector to which it belongs. In particular, a cluster is the locus of all nodes having the same coordinates in the above system.

IV. BASIC MODULAR ARITHMETIC

Since several derivations in this paper employ modular arithmetic, it is appropriate to offer the reader a quick refresher of the terminology and basic results used hereafter.

Given any two integers x and m , with $m \neq 0$, let $|x|_m$ denote the *modulo* operation, that is the nonnegative remainder of the division of x by m (see [20]). Two integers x and y are *congruent modulo* m , denoted by $x \equiv y \pmod{m}$, if and only if $|x|_m = |y|_m$.

Let \bullet indicate one of the three basic operations, *addition*, *subtraction*, and *multiplication*. The modulo operation distributes over such operations and hence $|x \bullet y|_m = ||x|_m \bullet |y|_m|_m = |x \bullet |y|_m|_m = ||x|_m \bullet |y|_m|_m$. Moreover, it is easy to prove that:

Property IV.1. For any integers a , x , and m , with $a \neq 0$ and $m \neq 0$, $|ax|_{am} = a|x|_m$.

Let the *greatest common divisor* of integers x and y be denoted by (x, y) . Therefore, letting $x = x'(x, m)$, and $m = m'(x, m)$ and applying Property IV.1, one derives $|x|_m = (x, m)|x'|_{m'}$. It is worth noting that the division of x by y modulo m is possible only when y and m are *coprime*, i.e. when $(m, y) = 1$. Indeed only in such a case there exists the *inverse multiplicative of y modulo m* , which is denoted by $|\frac{1}{y}|_m$ (as used in [20]) and is defined as that integer satisfying $x|\frac{1}{x}|_m \equiv 1 \pmod{m}$. The following property is widely used in the paper:

Property IV.2. Given any integers x, y, z, w , and $m \neq 0$, it holds:

- 1) If $ax \equiv ay \pmod{m}$ and $a \neq 0$ is such that $(a, m) = 1$, then $x \equiv y \pmod{m}$
- 2) If $ax \equiv ay \pmod{m}$ and $(a, m) = g$, then $x \equiv y \pmod{m'}$, where $m = m'g$

The next property shows how the values generated by the expression $|ix|_m$ vary when i assumes any integer value.

```

Procedure Actor ( $k, \tau_1$ );
  for  $z := 0$  to  $\tau_1 - 1$  do
    transmit the beacon  $|k - 1 - z|_k$  up to corona  $C_{|k-1-z|_k}$ ;

```

Fig. 2. The corona training protocol for the actor.

Property IV.3. Given two integers x and $m \neq 0$ such that $(x, m) = g$, the congruence $ix \equiv y \pmod{m}$ has solution for any $y = gy'$ with $y' \in [0, \dots, m' - 1]$, where $m' = \frac{m}{g}$. Moreover, $|ix|_m$ generates only the values multiple of g in $[0, 1, \dots, m - 1]$, one for each different value of $|i|_{m'}$.

In the particular case where $(x, m) = 1$, the property above shows that, when i assumes all the m integer values in $[0, \dots, m - 1]$, the expression $|ix|_m$ generates all the m integer values in $[0, \dots, m - 1]$.

V. THE FLAT- PROTOCOL

The main goal of this section is to present the details of the basic training protocol where each individual sensor has to learn the identity of the corona and of the sector to which it belongs, regardless of the moment when it wakes up for the first time.

The protocol acts as follows. Consider first the corona training task in which the actor broadcasts using the isotropic antenna. Immediately after deployment the actor 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. Specifically, the actor 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 actor 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 actor continues to transmit at decreasing power levels until it concludes its transmission cycle with a broadcast of beacon 0 that can be received only by the sensors in corona C_0 . In general, at time slot z , with $z \geq 0$, the actor transmits the beacon $k - 1 - |z|_k$ at a power level sufficient to cover the distance $r_{k-|z|_k}$ and hence to reach all the sensors up to corona $C_{k-1-|z|_k}$, but not those beyond $C_{k-1-|z|_k}$. The actor transmission cycle is repeated for a given time τ_1 which is sufficient to accomplish the entire corona training protocol (the actor can derive the value of τ_1 from k, L , and d , as it will be shown in the discussion after Theorem V.9). The protocol for the actor is shown in Figure 2.

The sector training task is analogous to the corona training task, except that now the actor broadcasts using the directional antenna. Indeed, the actor cyclically repeats a transmission cycle of h directional broadcasts with successively smaller angles. Specifically, at time slot z , with $z > 0$, the actor transmits the beacon $h - 1 - |z|_h$ which can reach all the sensors up to sector $S_{h-1-|z|_h}$, namely, using an angle of transmission $\alpha = (h - |z|_h) \frac{2\pi}{h}$. Since sector training is the same as corona training once the directional broadcasts replace the isotropic ones and h replaces k , all the results that will be presented for coronas hold also for sectors. Therefore, sector training will not be further discussed and we shall concentrate only on corona training for sensors.

In order to describe the basic corona protocol for sensors, called *Flat-*, we assume that each sensor is aware of the actor behaviour and of the total number k of coronas. In particular, k can be either stored in the sensor memory before deployment

```

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

```

Fig. 3. The *Flat-* protocol for a sensor.

or transmitted by the actor in the beacon along with the corona identity. 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 actor 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 the sensor learns 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 actor while the sensor is awake. As soon as the sensor hears an actor transmission for the first time, it starts to fill its register R and it is able to learn the actor global time t within the current actor transmission cycle, that is $t = |z|_k$. 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 actor 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 an awake sensor in corona C_c hears any broadcast which transmits beacon b , with $c \leq b \leq k - 1$ (clearly, different beacons are heard in different broadcasts). In contrast, an awake sensor in corona C_c cannot hear the broadcasts transmitting beacon b , with $0 \leq b \leq c - 1$, because it is out of the range covered by the actor transmission. 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 C_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 V.1. [23] *A sensor which belongs to corona C_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 C_0 is trained as soon as R_0 is set to 1.*

In the resulting sensor protocol, called *Flat-*, each sensor counts in ν the number of its sleep/wake transitions needed to be trained (line 1), initializing its local time when the sensor receives a

beacon for the first time from the actor (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 V.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.

A. Correctness and performance analysis

In the following, the conditions on the k, L , and d parameters 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.

Lemma V.2. *Given L, d , and k , there are exactly $k' = \frac{k}{(L,k)}$ different corona identities that can be transmitted by the actor 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 beacon transmitted when the sensor starts its i -th awake period is $|K_x - iL|_k = |K_x - i(L, k)|_{k'}|_{k'}$, where K_x is the corona identity transmitted at time x , that is $K_x = C_{|k-1-x|_k}$. Overall only k' different coronas can be transmitted by the actor when the sensor starts its awake periods, independent of how long the training process will be. Such k' coronas identities can be reindexed as $|K_x - s(L, k)|_k$, for $0 \leq s \leq k' - 1$.*

Proof: Consider a sensor that wakes up for the first time at the global time slot $z = x$, while the actor is transmitting the beacon $K_x = |k-1-x|_k = |k-1-x|_k$. The i -th sleep-awake cycle of such a sensor starts at time $x+iL$ while the actor is transmitting the beacon $|k-1-x-iL|_k = |K_x - iL|_k|_{k'}$, with $i \geq 0$. Observe that L and k can be rewritten as $L = gL'$ and $k = gk'$, where $g = (L, k)$. By Property IV.3, $|iL|_k$ generates only the k' multiple of g , one for each different value assumed by $i \bmod k'$, in $[0, \dots, k]$. Moreover, $K_x - iL \equiv K_x - g(|i|_{k'})|L'|_{k'} \pmod{k}$. In other words, in any two awake periods, say the i -th and the j -th ones, such that $i > j$ and $i - j < k'$, the coronas C_{x+iL} and C_{x+jL} are distinct and differ by a multiple of g . Whereas, in any two awake periods i and j such that $i \equiv j \pmod{k'}$ the same coronas are transmitted. Clearly, the k' different corona identities transmitted at the beginning of the awake periods 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 $|K_x - sg|_k$, with $0 \leq s \leq k' - 1$, corresponds to the first beacon transmitted in the j -th awake period, with $j = \left\lfloor s \frac{1}{L'} |k'|_{k'} \right\rfloor$. \square

Therefore, after exactly k' sleep-awake cycles, that is after $k'L$ time slots, which correspond to $\frac{k'L}{k} = \frac{k'L}{gk'} = \frac{L}{g} = L'$ actor transmission cycles, the behaviour of the sensor and the actor will be cyclically repeated. In other words, at the beginning of the k' -th awake period, the sensor and the actor 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 actor at least once. Thus, we have the following natural consequence.

Lemma V.3. *Given 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 actor 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 actor. Hence, in the worst case, the sensor needs $k' = \frac{k}{(L,k)}$ awake periods to hear the actor for the first time and further $k' = \frac{k}{(L,k)}$ awake periods to fill R . \square

Clearly, if the training condition of Lemma V.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, regardless 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 V.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: To simplify notation, write $g = (L, k)$. For the sake of contradiction, suppose that all the sensors have been trained and let $d < g$. By Lemmas V.2 and V.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 V.1 will be never satisfied.

Conversely, if $d \geq g$, by Lemma V.2, in k' consecutive sleep-awake cycles, the beacons transmitted by the actor 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 actor during an awake period of the sensor. Hence, after having heard the actor 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

From Theorem V.4 and recalling that any sensor is trivially trained when $d \geq k$, one has that the Flat- protocol is correct for any value of d in the range $(L, k) \leq d \leq k$. Next, the protocol performance is refined within such a range finding stricter upper bounds on the maximum number of sleep-awake periods required for training. The performance analysis starts from two particular cases, namely, $d = (L, k)$ and $d = |L|_k$ (note that $|L|_k = (L, k)|L'|_{k'} \geq (L, k)$). More precisely, Lemmas V.5 and V.6 specify, for $d = (L, k)$ and $d = |L|_k$ respectively, in which period a sensor, that wakes up for the first time at slot x , is awake while the actor transmits an arbitrary beacon c .

Lemma V.5. *Let c be any corona identity and assume $d = (L, k)$. The actor 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 $i_{c,x} = \left\lfloor \left\lfloor \frac{|K_x - c|_k}{d} \right\rfloor \left| \frac{1}{L'} \right|_{k'} \right\rfloor$, $L' = \frac{L}{d}$, and $k' = \frac{k}{d}$.*

Proof: When the sensor wakes up at time x the actor

is transmitting the beacon K_x . Moreover, the beacon values decrease within the actor transmission cycle. Thus, beacon c will be transmitted, starting from K_x , during the j -th group of d consecutive corona identities such that $j = \lfloor \frac{K_x - c}{d} \rfloor$. 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 actor transmits $\lfloor K_x - \lfloor \frac{K_x - c}{d} \rfloor d \rfloor_k$ as the first beacon. Hence, by Lemma V.2, $i_{c,x}$ is derived by solving the equation $\lfloor K_x - i_{c,x}(L, k) \rfloor_{k'} = \lfloor K_x - \lfloor \frac{K_x - c}{d} \rfloor d \rfloor_k$. By Property IV.2, it follows $i_{c,x} = \lfloor \lfloor \frac{K_x - c}{d} \rfloor \lfloor \frac{1}{L'} \rfloor_{k'} \rfloor_{k'}$ because $d = (L, k)$. \square

Lemma V.6. *Let c be any corona identity and assume $d = |L|_k$. The actor transmits beacon c during the $i_{c,x}$ -th awake period of a sensor which wakes up for the first time at slot x , where $i_{c,x} = \lfloor \frac{|K_x - c|_k}{d} \rfloor$.*

Proof: The proof is similar to that of Lemma V.5. Only observe that now, since $d = |L|_k = (L, k) |L'|_{k'}$ by Property IV.1, $i_{c,x}$ is derived by solving the equation $\lfloor K_x - i_{c,x}(L, k) \rfloor_{k'} = \lfloor K_x - \lfloor \frac{|K_x - c|_k}{d} \rfloor d \rfloor_k$, and hence $i_{c,x} = \lfloor \frac{|K_x - c|_k}{d} \rfloor$. \square

The following two lemmas determine, for $d = (L, k)$ and $d = |L|_k$ respectively, in which awake period a sensor, belonging to corona c and waking up for the first time at slot x , satisfies for the first time the training condition. Indeed, a sensor belonging to corona c is trained when it has filled both R_c and R_{c-1} and the awake period when such entries are filled can be derived from Lemmas V.5 and V.6 as follows:

Lemma V.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 actor 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 $\lfloor K_x - i(L, k) \rfloor_{k'} = \lfloor K_x - (j + 1)d \rfloor_k$, where $j = \lfloor \frac{|K_x - c|_k}{d} \rfloor$. Solving the above equation, one has $i = \lfloor (j + 1) \lfloor \frac{1}{L'} \rfloor_{k'} \rfloor_{k'}$, and hence $\lfloor \frac{1}{L'} \rfloor_{k'}$ awake periods after $i_{c,x}$. \square

Lemma V.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 V.7. For $d = |L|_k$, 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 $\lfloor i(L, k) \rfloor_{k'} = \lfloor (j + 1)d \rfloor_k$, and hence $i = j + 1$, where $j = \lfloor \frac{|K_x - c|_k}{d} \rfloor$. \square

From Lemmas V.7 and V.8, one knows the number of sleep-awake periods required by any sensor in the network to be trained. Letting ν_{\max} be the maximum number of sleep-awake periods required by the sensor that will be trained as the last one in the network, the worst case performance for the Flat- protocol results as follows:

Theorem V.9. *Given L , d , and k , if $d < (L, k)$ then there are sensors that cannot be trained by the Flat- protocol; otherwise all the sensors are trained, and:*

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

Proof: When $(L, k) \leq d < |L|_k$, since by Lemma V.2 the k' coronas transmitted by the actor 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 VI.3, observing that $\lfloor \frac{|K_x - c|_k}{d} \rfloor$ varies between 0 and $\frac{k}{d} - 1$, one has $\nu_{\max} \leq \frac{k}{(L, k)} + \lfloor \frac{1}{L'} \rfloor_{k'}$, 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 V.8, $\nu_{\max} \leq \lfloor \frac{k}{|L|_k} \rfloor + 1$. Note that, when k is a multiple of $|L|_k$, $\nu_{\max} = \lfloor \frac{k}{|L|_k} \rfloor + 1$ only for those sensors that wake up for the first time while the actor 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 actor is transmitting corona $c - 1$ and which belong to corona c . \square

To summarize, given an arbitrary sensor, belonging to a generic corona c and whose first wake up is any time slot x , Lemmas V.5 – V.8 determine the exact number of sleep-awake periods, or number of transitions, that such a sensor requires to be trained when d assumes the two particular values (L, k) and $|L|_k$. Since the sensor network is trained when all its sensors are trained, the exact number of transitions ν_{\max} to accomplish the training protocol is given by the number of transitions ν required by the sensor that is trained as last in the network. On the other hand, when the values of d are different from the two above, since it is difficult to analytically derive the exact number ν of transitions required by each sensor to be trained, ν_{\max} cannot be computed and thus an upper bound for it, based on the values of ν_{\max} computed for the two particular cases $d = (L, k)$ and $d = |L|_k$, is provided in Theorem V.9. However, the exact value of ν_{\max} can be experimentally computed, as it will be done in Section VII.

The maximum number of transitions ν_{\max} impacts all the performance parameters. Indeed, it is easy to see that the overall sensor awake time ω_{\max} is equal to $\nu_{\max}d$ while the total time for training all the sensors is $\nu_{\max}L + k$, where the additive term k comes from the fact that a sensor may wake up at any time $x < k$. Finally, referring to the actor behaviour, it should be clear that the actor repeats its transmission cycle for τ_1 slots, where $\tau_1 = \nu_{\max}L + k$ is derived by choosing the suitable value of ν_{\max} according to Theorem V.9.

We now turn to the analysis of the average case performance of the Flat- protocol, where it is assumed that the sensor awake time x is a discrete random variable uniformly distributed in $[0, k - 1]$. Let N be the total number of sensors, let N_c be the number of sensors that belong to corona c and, among them, let $N_{c,x}$ be those that wake up for the first time at slot x , with $0 \leq c, x \leq k - 1$. Since x is uniformly distributed, $N_{c,x} = \frac{N_c}{k}$ and, clearly, $\sum_{c=0}^{k-1} N_c = N$. Letting ν_{avg} and ω_{avg} denote the average number of transitions and the average overall sensor awake time, respectively, one has the following result.

Theorem V.10. Given 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 $\nu_{\text{avg}} \leq \frac{k'+1}{2} + \frac{1}{(L, k)} \left\lfloor \frac{1}{L'} \right\rfloor_{k'}$, where $k' = \frac{k}{(L, k)}$ and $L' = \frac{L}{(L, k)}$;
- 2) If $|L|_k \leq d < k$, then $\nu_{\text{avg}} \leq \left(\left\lfloor \frac{k}{|L|_k} \right\rfloor + 1 \right) \left(\frac{1}{2} + \frac{|k|_{|L|_k} + 1}{k} \right) - \frac{1}{k}$;
- 3) If $d = k$, then $\nu_{\text{avg}} \leq 1 + \frac{1}{k}$.

Proof: In the following, the notation $\ell \leq x \leq r$ will denote either the values $\ell, \ell + 1, \dots, r - 1, r$ when $\ell \leq r$ or the values $\ell, \ell + 1, \dots, k - 1, 0, \dots, r - 1, r$ when $\ell > r$. Consider first the range $(L, k) \leq d < |L|_k$. Assume $d \geq 2$. Fixed a corona c , the sensors that are trained during the 0-th sleep-awake period are those that wake up at $|k - 1 - (c + d - 2)|_k \leq x \leq k - 1 - c$. Indeed, such sensors during the d slots of the first awake period hear the beacon c and set $R[c] = 1$ and $R[c - 1] = 0$. In contrast, the sensors that wake up at $x = |k - 1 - (c + d - 1)|_k$ can set $R[c]$ to 1, but cannot set $R[c - 1]$ to 0 until the $\left\lfloor \frac{1}{L'} \right\rfloor_{k'}$ -th sleep-awake period. In general, the sensors that wake up at $|k - 1 - (c + d - 2 + i(L, k)|_{L'}|_{k'})|_k \leq x \leq |k - 1 - (c + i((L, k)|_{L'}|_{k'}))|_k$ are trained during the i -th sleep-awake period because they set both $R[c]$ and $R[c - 1]$. In contrast, the sensors that wake up at $x = |k - 1 - (c + d - 1 + i(L, k)|_{L'}|_{k'})|_k$ will set $R[c - 1]$ at the $(i + \left\lfloor \frac{1}{L'} \right\rfloor_{k'})$ -th sleep-awake period. When $d = (L, k)$, the average number of transitions required to be trained by a sensor in corona c is bounded above by

$$\begin{aligned} \bar{\nu}_c &\leq \frac{1}{N_c} \left(\sum_{i=0}^{k'-1} \sum_{x=|k-1-(c+d-1+i(L,k)|_{L'}|_{k'})|_k}^{|k-1-(c+i(L,k)|_{L'}|_{k'})|_k} (i+1)N_{c,x} + \right. \\ &\quad \left. \sum_{i=0}^{k'-1} \left\lfloor \frac{1}{L'} \right\rfloor_{k'} N_{c,|k-1-(c+d-1+i(L,k)|_{L'}|_{k'})|_k} \right) \\ &= \frac{1}{N_c} \left(\sum_{i=1}^{k'} id \frac{N_c}{k} + k' \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \frac{N_c}{k} \right) \\ &= \frac{k'+1}{2} + \frac{1}{(L, k)} \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \end{aligned}$$

If $d = (L, k) = 1$, no sensor in corona c can be trained in a single awake period, regardless its first wake up time x , because it can hear at most one beacon. Thus, each sensor has to wait at least $1 + \left\lfloor \frac{1}{L} \right\rfloor_k$ awake periods in order to set both $R[c]$ and $R[c - 1]$. Therefore,

$$\begin{aligned} \bar{\nu}_c &\leq \frac{1}{N_c} \left(\sum_{i=0}^{k-1} \left\lfloor \frac{1}{L} \right\rfloor_k N_{c,|k-1-(c+i|L|_k)|_k} + \right. \\ &\quad \left. \sum_{i=0}^{k-1} (i+1)N_{c,|k-1-(c+i|L|_k)|_k} \right) \\ &= \frac{1}{N_c} \left(\sum_{i=0}^{k-1} \left\lfloor \frac{1}{L} \right\rfloor_k \frac{N_c}{k} + \sum_{i=0}^{k-1} (i+1) \frac{N_c}{k} \right) \\ &= \left\lfloor \frac{1}{L} \right\rfloor_k + \frac{k+1}{2} \end{aligned}$$

Since $\bar{\nu}_c$ does not depend on c , summing up over the coronas, $\nu_{\text{avg}} = \frac{1}{N} \sum_{c=0}^{k-1} \bar{\nu}_c N_c = \bar{\nu}_c$.

Consider now $1 \leq |L|_k \leq d < k$. Fixed a corona c , as before, the sensors that wake up at $|k - 1 - (c + d - 2)|_k \leq x \leq k - 1 - c$ are trained during the 0-th sleep-awake period. In contrast, the sensors that wake up at $x = |k - 1 - (c + d - 1)|_k$ are trained during the next sleep-awake period. In general, the sensors that wake up at $|k - 1 - (c + d - 2 + i|L|_k)|_k \leq x \leq |k - 1 - (c + (i + 1)d - 2)|_k \leq x \leq |k - 1 - (c + i|L|_k)|_k = |k - 1 - (c + id)|_k$ are trained during the i -th sleep-awake period, while those waking up at $x = |k - 1 - (c + (i + 1)d - 1)|_k$ are trained in the successive sleep-awake period. Therefore, the average number of transitions required to be trained by a sensor in corona c is

$$\begin{aligned} \bar{\nu}_c &= \frac{1}{N_c} \left(\sum_{x=|k-c-d+1|_k}^{k-1-c} N_{c,x} + \right. \\ &\quad \left. \sum_{i=1}^{\left\lfloor \frac{k}{|L|_k} \right\rfloor - 1} \sum_{x=|k-c-(i+1)d+1|_k}^{|k-c-id|_k} (i+1)N_{c,x} + \right. \\ &\quad \left. \sum_{x=|k-c-|k|_{|L|_k}|_k}^{k-c} \left(\left\lfloor \frac{k}{|L|_k} \right\rfloor + 1 \right) N_{c,x} \right) \end{aligned}$$

where the last sum is due to the sensors that are trained in the $\left\lfloor \frac{k}{|L|_k} \right\rfloor$ -th sleep-awake period. In such a sum, when $|k|_{|L|_k} = 0$, only the sensors that belong to corona c and wake up at time slot $k - c$, while the actor is transmitting the beacon $c - 1$, have to wait $\frac{k}{|L|_k} + 1$ sleep-awake periods to be trained. With simple algebraic manipulations, when $d = |L|_k$, one gets:

$$\begin{aligned} \bar{\nu}_c &\leq \frac{1}{N_c} \left((d-1) \frac{N_c}{k} + \sum_{i=2}^{\left\lfloor \frac{k}{|L|_k} \right\rfloor} id \frac{N_c}{k} + \right. \\ &\quad \left. \left(\left\lfloor \frac{k}{|L|_k} \right\rfloor + 1 \right) \left(|k|_{|L|_k} + 1 \right) \frac{N_c}{k} \right) \\ &= \left(\left\lfloor \frac{k}{|L|_k} \right\rfloor + 1 \right) \left(\frac{1}{2} + \frac{|k|_{|L|_k} + 1}{k} \right) - \frac{1}{k} \end{aligned}$$

As before, summing up over all coronas, one has $\nu_{\text{avg}} = \bar{\nu}_c$.

When $d = k$, only the sensors that belong to corona c and wake up at time slot $x = k - c$ are not trained in a single awake period, but they require one more period. Hence:

$$\nu_{\text{avg}} = 1 + \frac{1}{N} \sum_{c=0}^{k-1} N_{c,c-1} = 1 + \frac{1}{N} \sum_{c=0}^{k-1} \frac{N_c}{k} = 1 + \frac{1}{k}$$

As observed in the proof of Theorem IV.9, when either $(L, k) < d < |L|_k$ or $|L|_k < d < k$, more sensors can be trained in each sleep-awake period, and hence ν_{avg} cannot be greater than that for $d = (L, k)$ and $d = |L|_k$, respectively. \square

It is worthy to point out that, as shown by Theorems V.9 and V.10, the performance of the Flat- protocol depends on the values of k , d , and L . Therefore, it is conceivable that in mission critical systems these parameters could be tuned before the sensor deployment in order to guarantee a predefined performance quality.

VI. IMPROVEMENTS

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

```

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  $R_{|c+j|_k+i-h|_k} := 0$ ;

```

Fig. 4. The extra instructions for the Flat protocol.

A. The Flat protocol

As a first improvement to Flat-, recall that, as soon as a sensor hears the actor transmission for the first time, it learns from the beacon the actor global time modulo the actor transmission cycle. Therefore, it can immediately retrieve backwards the coronas which it did not hear and which were transmitted by the actor during its previous awake periods, setting to 0 the corresponding entries of R . The resulting improved protocol, called *Flat*, is derived from the Flat- protocol by modifying, as shown in Figure 4, the `if` instruction in lines 6-7 of Figure 3. As a drawback, a sensor may now execute as many as $O(\nu_{\max}d)$ 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 actor for the first time, it fills R as it would have heard the actor since the first time it woke up, Lemma V.3 and Theorem V.4 can be restated as follows:

Lemma VI.1. *Given 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.*

Theorem VI.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 V.5 and V.6 still hold, Lemmas V.7 and V.8 can be restated for the Flat protocol as follows:

Lemma VI.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 V.7 and V.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 VI.4. *Given L, d , and k , if $d < (L, k)$ then there exist sensors that cannot be trained by the Flat protocol; otherwise all the sensors are trained, and:*

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

Proof: When $d = (L, k)$, by Lemmas V.5 and VI.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 V.6 and VI.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 V.2 the k' coronas transmitted by the actor 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 actor is transmitting corona $c-1$ will set R_{c-1} the first time they hear the actor, without waiting that the actor 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_{\max} matches the upper bound given in Theorem VI.4.

With respect to the average case performance of the Flat protocol, one can prove the following result.

Theorem VI.5. *Given 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 $\nu_{\text{avg}} \leq \frac{k'+1}{2} + \frac{1}{(L,k)} \left\lfloor \frac{1}{L'} \right\rfloor_{k'} - \frac{1}{2k} \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \left(\left\lfloor \frac{1}{L'} \right\rfloor_{k'} + 1 \right)$, where $k' = \frac{k}{(L,k)}$ and $L' = \frac{L}{(L,k)}$;
- 2) If $|L|_k \leq d < k$, then $\nu_{\text{avg}} \leq \left(\left\lfloor \frac{k}{|L|_k} \right\rfloor + 1 \right) \left(\frac{1}{2} + \frac{|k||L|_k+1}{k} \right) - \frac{1}{k}$;
- 3) If $d = k$, then $\nu_{\text{avg}} = 1$.

Proof: The proof is similar to that of Theorem V.10. Consider first the range $(L, k) \leq d < |L|_k$ with $d \geq 2$. From Theorem V.4, $\nu_{\max} \leq k'$ for the Flat protocol. Hence, the sensor behaviour is the same as that for the Flat- protocol, except for those sensors which were trained by Flat- using more than k' transitions. Therefore, the number of transitions remains the same except for the sensors that wake up at slot $x = |k-1 - (c+d-1+i(L,k)|L'|_{k'})|_k$, with $k' - \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \leq i \leq k' - 1$, which will now set $R[c-1]$ no later than during the $(k'-1)$ -th sleep-awake period. Thus, the formula for $\bar{\nu}_c$ becomes

$$\begin{aligned}
\bar{\nu}_c &\leq \frac{1}{N_c} \left(\sum_{x=|k-1-(c+d-2)|_k}^{k-1-c} N_{c,x} + \right. \\
&\quad \left. \left(1 + \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \right) N_{c,|k-1-(c+d-1)|_k} + \right. \\
&\quad \left. \sum_{i=1}^{k'-1} \sum_{x=|k-1-(c+d-2+i(L,k)|L'|_{k'})|_k}^{k-1-(c+i(L,k)|L'|_{k'})|_k} (i+1) N_{c,x} + \right. \\
&\quad \left. \sum_{i=1}^{k'-1-\left\lfloor \frac{1}{L'} \right\rfloor_{k'}} \left(i+1 + \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \right) N_{c,|k-1-(c+d-1+i(L,k)|L'|_{k'})|_k} + \right. \\
&\quad \left. \sum_{i=k'-\left\lfloor \frac{1}{L'} \right\rfloor_{k'}}^{k'-1} k' N_{c,|k-1-(c+d-1+i(L,k)|L'|_{k'})|_k} \right) \\
&= \frac{k'+1}{2} + \frac{1}{(L,k)} \left\lfloor \frac{1}{L'} \right\rfloor_{k'} - \frac{1}{2k} \left\lfloor \frac{1}{L'} \right\rfloor_{k'} \left(\left\lfloor \frac{1}{L'} \right\rfloor_{k'} + 1 \right)
\end{aligned}$$

If $d = (L, k) = 1$, since no sensor in corona c can be trained in a single awake period and $\nu_{\max} \leq k' = k$ by Theorem V.4,

one can write:

$$\begin{aligned}
\bar{\nu}_c &\leq \frac{1}{N_c} \left(\left(1 + \left\lfloor \frac{1}{L} \right\rfloor_k \right) N_{c, |k-1-c|_k} + \right. \\
&\quad \left. \sum_{i=1}^{k-1-\lfloor \frac{1}{L} \rfloor_k} \left(i + 1 + \left\lfloor \frac{1}{L} \right\rfloor_k \right) N_{c, |k-1-(c+i|L|_k)|_k} + \right. \\
&\quad \left. \sum_{i=k-\lfloor \frac{1}{L} \rfloor_k}^{k-1} k N_{c, |k-1-(c+i|L|_k)|_k} \right) \\
&= \frac{1}{N_c} \left(\sum_{i=0}^{k-1-\lfloor \frac{1}{L} \rfloor_k} \left(i + 1 + \left\lfloor \frac{1}{L} \right\rfloor_k \right) \frac{N_c}{k} + \sum_{i=k-\lfloor \frac{1}{L} \rfloor_k}^{k-1} k \frac{N_c}{k} \right) \\
&= \frac{1}{N_c} \left(\frac{N_c}{k} \sum_{i=1+\lfloor \frac{1}{L} \rfloor_k}^k i + \left\lfloor \frac{1}{L} \right\rfloor_k N_c \right) \\
&= \frac{k+1}{2} + \left\lfloor \frac{1}{L} \right\rfloor_k - \frac{1}{2k} \left\lfloor \frac{1}{L} \right\rfloor_k \left(\left\lfloor \frac{1}{L} \right\rfloor_k + 1 \right)
\end{aligned}$$

Summing over all the coronas, one has $\nu_{\text{avg}} = \frac{1}{N} \sum_{c=0}^{k-1} \bar{\nu}_c N_c = \bar{\nu}_c$.

Consider now the case $1 \leq |L|_k \leq d < k$. The Flat protocol has the same behaviour as the Flat- one, except in the particular case that $d = |L|_k$ divides k . In such a case, the sensors that belong to corona c and wake up for the first time when the actor transmits the beacon $c-1$ save one transition. Indeed, such sensors set $R[c-1]$ the first time they hear any beacon, thus saving $\frac{1}{k}$ on the average number of transitions. However, in general, the same bound of ν_{avg} stated in Theorem IV.10 still holds.

Finally, when $d = k$, each sensor is trained in a single transition, and hence $\nu_{\text{avg}} = \nu_{\text{max}} = 1$. \square

B. 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 actor 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 it finds a future awake period whose corresponding d entries are not already filled. The resulting protocol, called Flat+, is illustrated in Figure 5. 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 12). 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 20). When the sensor hears the actor for the first time, it stores in max0 the largest entry of R which must be 0 due to its previous awake periods (line 8), and thus it sets to 0 the entries from R_0 to R_{max0} (lines 9–10). Finally, at the end of

Procedure Flat+ (k, L, d);

```

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

```

Fig. 5. The Flat+ protocol for a sensor.

the awake period, the sensor performs the above mentioned look-ahead process, properly setting its alarm-clock (lines 25–32).

The worst case performance of the Flat+ protocol coincides with that of the Flat one, as stated below.

Theorem VI.6. *Given 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 $\nu_{\text{max}} \leq \frac{k}{(L, k)}$;
- 2) If $|L|_k \leq d < k$, then $\nu_{\text{max}} \leq \left\lceil \frac{k}{|L|_k} \right\rceil$;
- 3) If $d = k$, then $\nu_{\text{max}} = 1$.

Proof: Clearly, the number of sleep/wake transitions of Flat+ cannot be larger than that of Flat. In fact, when $d = (L, k)$ or $d = |L|_k$, one can find bad instances, where ν_{max} is the same for both Flat+ and Flat. When $d = (L, k)$, consider a sensor that belongs to corona $c = k - 1$ and wakes up for the first time at $x = |d|L'|_{k'}|_k$, where, as usual, $k' = \frac{k}{(L, k)}$ and $L' = \frac{L}{(L, k)}$. Indeed, the sensor can hear only the beacon $k - 1$ which is, according to Lemma V.5, actually heard at the $(k' - 1)$ -th sleep-awake cycle. When $d = |L|_k$, a sensor which belongs to corona c and wakes up at time slot $x = k - c$, that is when the actor transmits $c - 1$, has to wait until the actor transmits beacon c , which happens at the $(\lceil \frac{k}{|L|_k} \rceil - 1)$ -th sleep-awake cycle. Thus, in both cases, the sensor cannot save transitions by means of its look-ahead process. \square

Besides to the worst case performance, also the average case performance of both the Flat and Flat+ protocols coincide when $(L, k) = 1$, for d ranging in $[|L|_k, k]$, because during subsequent awake periods of the sensors the actor transmits consecutive

```

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

```

Fig. 6. *The two-level protocol for the actor.*

decreasing beacons, and hence Flat+ cannot take advantage of its look-ahead process. Similarly, when $(L, k) \neq 1$ and d is sufficiently large, both Flat+ and Flat have the same behaviour. Indeed, for $d \geq \max\{|L|_k + 1, k - |L|_k\}$, both protocols require the same number of transitions, precisely one or two transitions, for each sensor. In fact, for such values of d , all the k beacons are transmitted by the actor during at most two sensor awake periods. So, since Flat+ cannot skip the first awake period, it cannot gain from its look-ahead process.

Although the average case performance of both the Flat and Flat+ protocols coincide for large values of d , or when $(L, k) = 1$ and $d \geq |L|_k$, the practical behaviour of Flat+ is much better than that of Flat for small values of d , as it will be experimentally checked in Section VII.

C. The two-level approach

The protocols discussed so far can be further improved by following a nested approach in which the k coronas are viewed as k_1 macrocoronas of k_2 adjacent coronas each. Specifically, 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 $0 \leq c \leq k_1k_2 - 1$. For determining both the macrocorona and microcorona identities, any of the Flat protocol variants can be used.

The protocol for the actor is shown in Figure 6. The actor works in two levels. In the first level, the actor 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 0, the actor 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_1-1} . At time slot 1, the actor transmits the beacon $k_1 - 2$ at a power that can be received up to the $(k_1 - 2)$ -th macrocorona, that is corona $C_{k_1-k_2-1}$. For the subsequent $k_1 - 2$ slots, the actor continues to transmit at decreasing powers until it concludes its cycle at time slot $k_1 - 1$ 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} . In general, at time slot z , with $0 \leq z \leq \tau_1 - 1$, the actor transmits the beacon $|k_1 - 1 - z|_{k_1}$ up to corona $C_{k_2(|k_1-1-z|_{k_1+1})-1}$, where 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 actor 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, choosing τ_2 so as to allow all the sensors in each macrocorona m to be also trained with respect to their microcorona. In

general, for each macrocorona $m = 0, 1, \dots, k_1 - 1$ and for each $z = 0, 1, \dots, \tau_2 - 1$, the actor transmits the beacon $|k_2 - 1 - z|_{k_2}$ up to corona $C_{mk_2+|k_2-1-z|_{k_2}}$. 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 actor 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+. The values of τ_1 and τ_2 are properly set according to the chosen variant, as stated in Theorems V.9, VI.4, and VI.6, respectively. As regard to the performance of the two-level protocols, one has:

Theorem VI.7. *Given 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. \square*

Note that, by Theorems V.9 and VI.4, tight bounds on the values of τ_1 and τ_2 can be derived only when $d = (L, k)$, $d = |L|_k$, and $d = k$. In all other cases, the total time of each level is derived from $\tau = \nu_{\max}L + k$, setting ν_{\max} equal to the upper bound given in Theorems V.9 and VI.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 $\min\{k_1, k_2\}$. When $d = (L, k)$, by Lemma VI.1 and Theorem VI.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 $\frac{k_1}{(L, k_1)} + \frac{k_2}{(L, k_2)} < \frac{k_1}{(L, k_1)} \frac{k_2}{(L, k_2)} = \frac{k}{(L, k_1)(L, k_2)} = \frac{k}{d}$. 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.

VII. EXPERIMENTAL TESTS

The worst and average performance of the corona training protocols were 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 of radius $\rho = k$, centered at the actor and inscribed in a square. Precisely, the Cartesian coordinates of each sensor are uniformly generated choosing at random two real numbers in the range $[-k, k]$. The generation proceeds until N sensors are placed inside the circle, thus discarding those laying outside.

In the experiments, fixed k , L , and N , and varying d between (L, k) and k , both the worst and average number of transitions, denoted by ν_{\max} and ν_{avg} , both the worst and average overall sensor awake time, $\omega_{\max} = \nu_{\max}d$ and $\omega_{\text{avg}} = \nu_{\text{avg}}d$, and the total time τ , which measures the time required to terminate the whole training process, are evaluated. The ν_{\max} value is obtained by taking the maximum ν among all sensors in the network, while ν_{avg} is computed by summing up the ν values of all sensors in the network and then dividing by N . The results are averaged over 3 independent experiments, which only differ in the deployment distribution of the sensors and in the sensor first wakeup times. It is worth noting that, if the network is dense enough to guarantee that there is at least one sensor in each corona for each first wakeup time, then ν_{\max} is always the same in different experiments, while ν_{avg} may slightly change depending on the sensor first wakeup time distribution.

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. Although L can assume any value, larger values are preferred in order to increase the longevity of the wireless sensor network. In fact, fixed d , a larger 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. Thus, in all our experiments, $L \geq k$ has been chosen. 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 $d \geq 8$.

Figure 7 shows the number ν_{\max} and ν_{avg} of transitions for the different values of d . According to Theorems V.9 and VI.4, when $d = 8$, Flat- has $\nu_{\max} = \frac{k}{(L, k)} + \left\lfloor \frac{1}{L'} \right\rfloor_{k'} = 8 + 5 = 13$, while both Flat and Flat+ have $\nu_{\max} = \frac{k}{(L, k)} = 8$. 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-. For $d \geq \max\{|L|_k + 1, k - |L|_k\}$, both the worst and average results of Flat and Flat+ coincide.

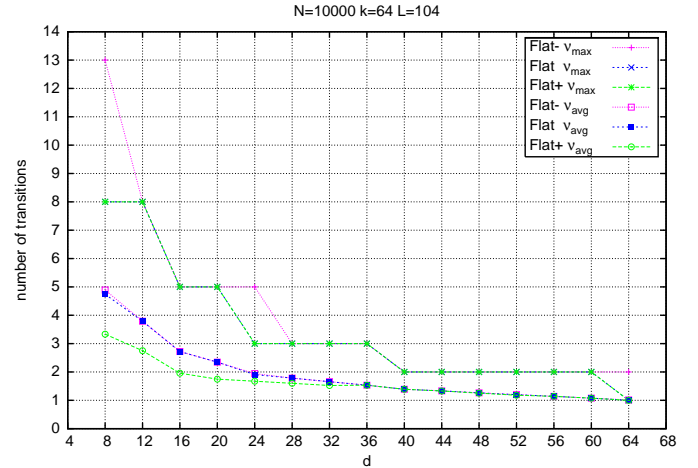


Fig. 7. Number of transitions when $k = 64$, $L = 104$, and $8 \leq d \leq 64$.

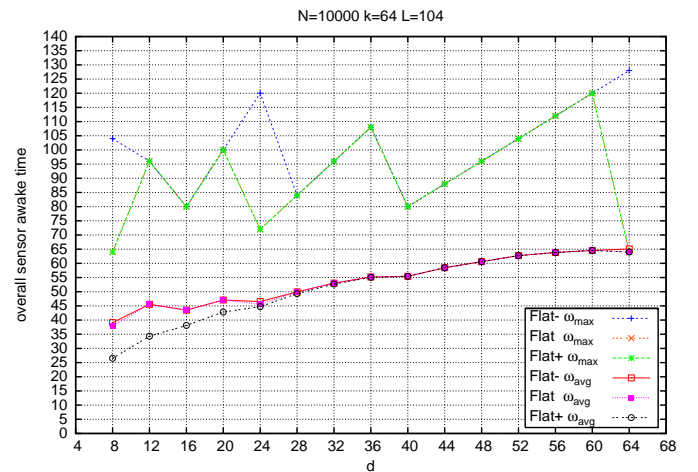


Fig. 8. Overall sensor awake time when $k = 64$, $L = 104$, and $8 \leq d \leq 64$.

Figure 8 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 8 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 V.9 and VI.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 actor for 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 9 illustrates 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 V.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 7 and 8. 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 9. 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 .

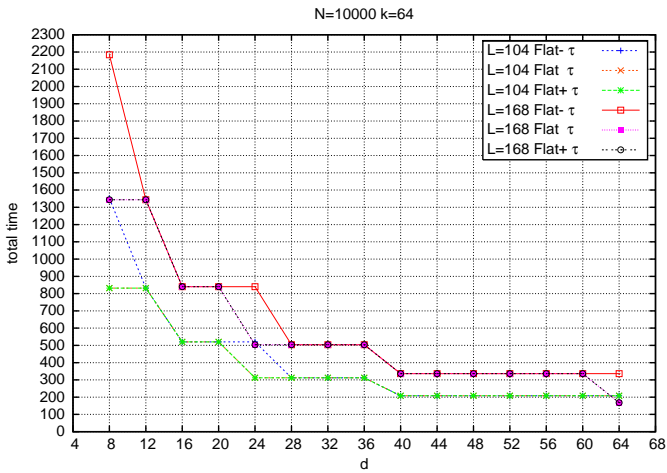


Fig. 9. Total time for training when $k = 64$, $L = 104$ or $L = 168$, and $8 \leq d \leq 64$.

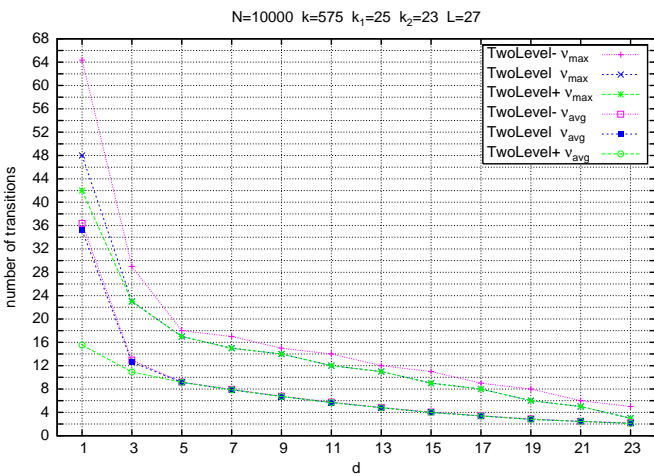


Fig. 10. Number of transitions when $k = 575$, $k_1 = 25$, $k_2 = 23$, $L = 27$, and $1 \leq d \leq 23$.

Consider now the experiments relative to the two-level approach. Recall that TwoLevel-, 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 $\min\{k_1, k_2\} = 23$.

Figures 10, 11, and 12 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 10 from the worst case performance of the one-level protocols. For example, when $d = (L, k_1) = (L, k_2) = 1$, TwoLevel- requires $\nu = 67$ sleep/wake transitions because the Flat- protocol requires $\nu_1 = \frac{k_1}{(L, k_1)} + \left\lfloor \frac{1}{L} \right\rfloor_{k_1} = 25 + 13 = 38$ transitions when $k_1 = 25$ and $\nu_2 = \frac{k_2}{(L, k_2)} + \left\lfloor \frac{1}{L} \right\rfloor_{k_2} = 23 + 6 = 29$ transitions when $k_2 = 23$.

Figure 11 shows the awake times $\omega_{\max} = \nu_{\max}d$ and $\omega_{\text{avg}} = \nu_{\text{avg}}d$. The curves in Figure 11 smoothly change, without the abrupt peaks of Figure 8, because now the number of transitions monotonically decreases, as shown in Figure 10. Moreover, note

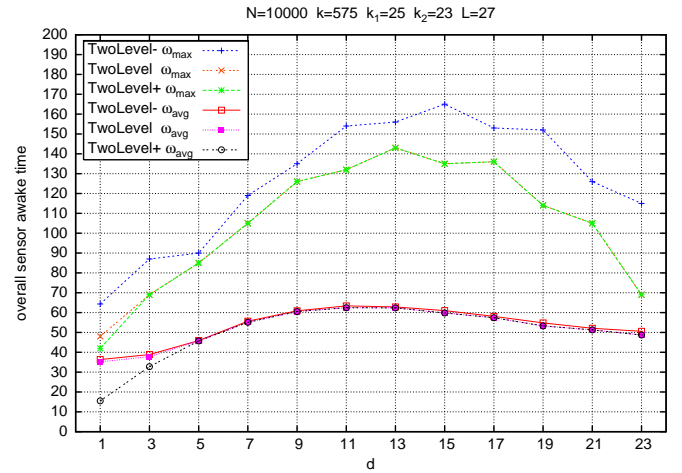


Fig. 11. Overall sensor awake time when $k = 575$, $k_1 = 25$, $k_2 = 23$, $L = 27$, and $1 \leq d \leq 23$.

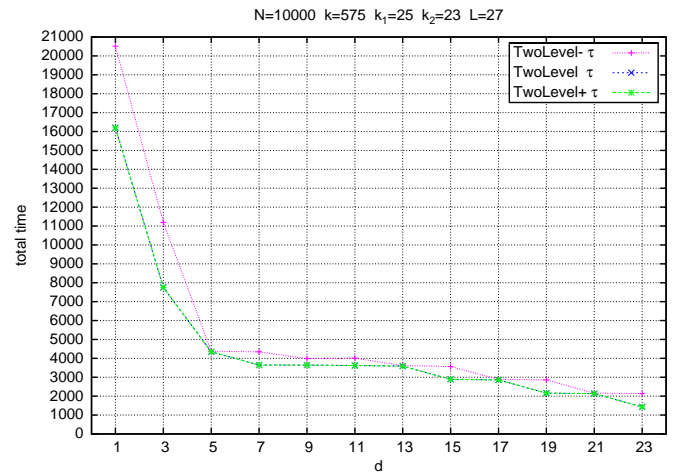


Fig. 12. Total time for training when $k = 575$, $k_1 = 25$, $k_2 = 23$, $L = 27$, and $1 \leq d \leq 23$.

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.

Figures 13, 14, and 15 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. The sensor awake period d varies, with a step of 5, between $(L, k) = 1$ and 76.

As expected for $1 = (L, k) \leq d \leq \min\{k_1, k_2\} = 23$, 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 .

VIII. 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

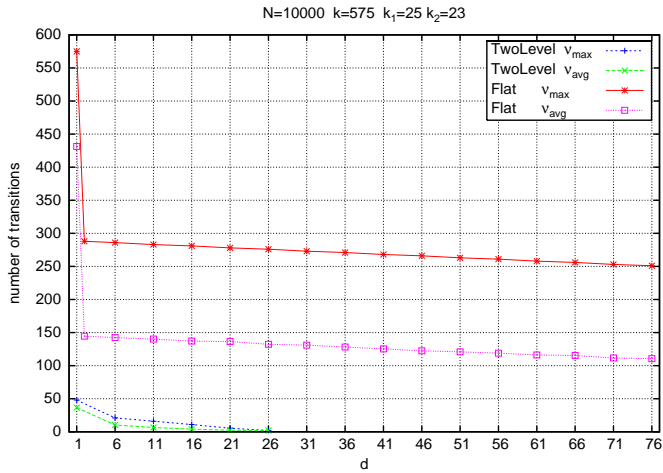


Fig. 13. Comparing the number of transitions between Flat and TwoLevel.

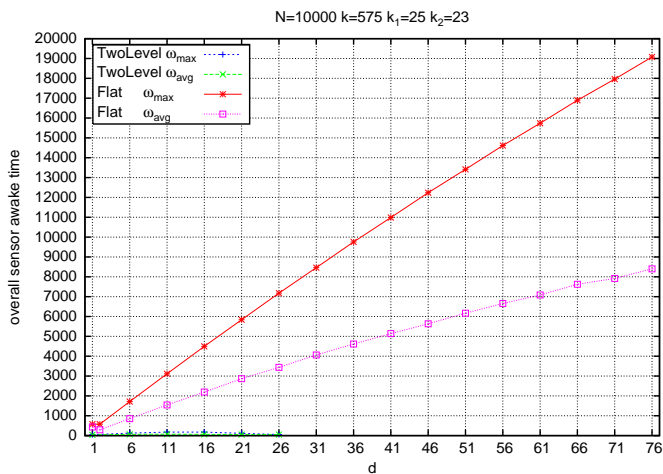


Fig. 14. Comparing the overall sensor awake time between Flat and TwoLevel.

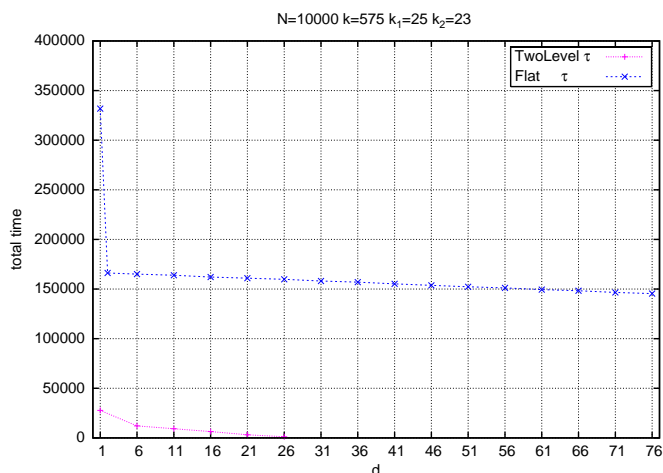


Fig. 15. Comparing the total time for training between Flat and TwoLevel.

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 .

As possible issues for further research, one could either generalize the two-level approach to a larger number of levels, finding whether there are numbers of levels for which optimal performances are attained, or search for significant lower bounds on such optimal performances. However, one should remind that some of the performance measures, such as the total training delay and the energy consumption per sensor, are orthogonal. Therefore, one should look for a good trade-off, especially considering the values that the parameters can assume in a realistic scenario.

In this paper, a single fixed actor was assumed for training the sensors in an ideal environment where the radio antennae transmit in a regular way. In practice, there is a variance in the received signal strength, which might exhibit some continuous changes in range and angle. Thus, one could investigate the accuracy of the acquired coarse-grained location, for example, by modeling the radio irregularity as done in [26]. Moreover, one could extend the protocols considering the case where there are several fixed actors, knowing about each other, and thus determine whether they can collaboratively train sensors better than using only a single actor. Finally, one could study how the protocols presented in this paper can be adapted to the case of one or more mobile actors.

ACKNOWLEDGMENT

Prof. Olariu's work was supported in part by NSF grants CNS-0721563 and CNS-0721586.

REFERENCES

- [1] I.F. Akyildiz and I. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2, 351–367, 2004.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.
- [3] J. Bachrach and C. Taylor. *Handbook of Sensor Networks*. Wiley, 2005.
- [4] S. Bandyopadhyay and E. Coyle. An efficient hierarchical clustering algorithm for wireless sensor networks. *Proc. IEEE INFOCOM 2003*, San Francisco, CA, April 2003.
- [5] A. A. Bertossi, S. Olariu, and M.C. Pinotti. Efficient corona training protocols for sensor networks. *Theoretical Computer Science*, 402 (1):2–15, 2008.
- [6] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran. Self-configuring localization systems: Design and experimental evaluation. *ACM Transactions on Embedded Computing Systems (TECS)*, 3(1):24–60, 2004.
- [7] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *IEEE Computer*, 37(8):41–49, 2004.
- [8] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal energy-aware clustering in sensor networks. *Sensors*, 2:258–269, 2002.
- [9] S. Giordano, S. Olariu, and D. Simplot-Ryl (Eds.), *SANET '07: Proceedings of the First ACM Workshop on Sensor and Actor Networks*, ACM, New York, 2007.
- [10] T. He, C. Huang, B. M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. In *MobiCom '03*, San Diego, CA, September 2003.
- [11] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer-Verlag, 1993.

- [12] K. Langendoen and N. Reijers. Distributed localization algorithm. In *Embedded Systems Handbook*, R. Zurawski (Editor), CRC Press, Boca Raton, FL, 2004.
- [13] D. Nicolescu. Positioning in ad-hoc sensor networks. *IEEE Network*, 18(4):24–29, 2004.
- [14] S. Olariu, M. Eltoweissy, and M. Younis. ANSWER: autonomous networks sensor systems. *Journal of Parallel and Distributed Computing*, 67, 114–126, 2007.
- [15] S. Olariu, A. Waada, L. Wilson, and M. Eltoweissy. Wireless sensor networks leveraging the virtual infrastructure. *IEEE Network*, 18(4):51–56, 2004.
- [16] S. Roundy, P. K. Wright, and J. Rabaey. *Energy Scavenging for Wireless Sensor Networks with Special Focus on Vibrations*. Kluwer Academic Press, Norwell, 2004.
- [17] M. Rudafshani and S. Datta, Localization in Wireless Sensor Networks. In *IPSN'07*, Cambridge, MA, April 2007.
- [18] A. Savvides, L. Girod, M. Srivastava, and D. Estrin. Localization in Sensor Network. In *Wireless Sensor Networks*, C.S. Raghavendra, K.M. Sivalingam, and T. Znati, Eds., Kluwer Academic, 2004.
- [19] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5), 16–27, 2000.
- [20] N. S. Szabo and R. I. Tanaka. *Residue Arithmetic and its Applications to Computer Technology*. McGraw-Hill, New York, 1967.
- [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] F. Ye, H. Zhang, S. Lu, L. Zhang, and J. Hou. A randomized energy-conservation protocol for resilient sensor networks. *Wireless Networks*, 12(5):637–652, 2006.
- [25] V.V. Zhirnov and D.J.C. Herr. New frontiers: self-assembly and nano-electronics. *IEEE Computer*, 34(1):34–43, 2001.
- [26] G. Zhou, T. He, S. Krishnamurthy, and J.A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, 2006.