Efficient Binary Corona Training Protocols for Heterogeneous Sensor and Actor Networks

F. Barsi, A.A. Bertossi, C. Lavault, A. Navarra, S. Olariu, M.C. Pinotti, and V. Ravelomanana

***Abstract*—Sensor networks are expected to evolve into long-lived, autonomous networked systems whose main mission is to provide in-situ users – called *actors* – with real-time information for specific goals supportive of their mission. The network is populated with a heterogeneous set of tiny sensors. The *free* sensors alternate between sleep and awake periods, under program control in response to computational and communication needs. The *periodic* sensors alternate between sleep periods and awake periods of predefined lengths, established at the fabrication time. The architectural model of an *actor-centric* network used in this work comprises in addition to the tiny sensors a set of mobile actors that organize and manage the sensors in their vicinity. We take the view that the sensors deployed are *anonymous* and unaware of their geographic location. Importantly, the sensors are not, a priori, organized into a network. It is, indeed, the interaction between the actors and the sensor population that organizes the sensors in a disk around each actor into a short-lived, mission-specific, network that exists for the purpose of serving the actor and that will be disbanded when the interaction terminates. The task of setting up this form of actor-centric network involves a *training* stage where the sensors acquire dynamic coordinates relative to the actor in their vicinity. The main contribution of this work is to propose an energy-efficient training protocol for actor-centric heterogeneous sensor networks. Our protocol outperforms all known training protocols in the number of sleep/awake transitions per sensor needed by the training task. Specifically, in the presence of k coronas, no sensor will experience more than $1+\lceil \log k \rceil$ sleep/awake transitions and awake periods.**

***Index Terms*—Autonomous wireless sensor networks, heterogeneous sensor and actor networks, free sensors, periodic sensors, training protocols**

I. INTRODUCTION

We assume a large-scale, random deployment of micro-sensors, each perhaps no larger than a dime, and possessing only limited functionality. The sensors are organized, under the control of an actor, into a short-lived, service-centric and mission-driven network. This view is in sharp departure from the common understanding that sensor networks are deployed in support of a remote entity that is querying the network and where the collected data is sent to a remote site for processing. In an *actor-centric* network the concept of globality has been redefined to mean small-scale spatial and temporal globality, the only viable form of non-local interaction. No global aggregation or fusion of sensory data is performed because such operations do not scale well with the size of the deployment area. Actor-centric sensor networks can detect trends and unexpected, coherent, and emergent behaviors and find immediate applications to homeland security [5], [8].

F. Barsi, A. Navarra, and M.C. Pinotti are with the Department of Computer Science and Mathematics, University of Perugia, 06123 Perugia, Italy, {barsi,pinotti}@unipg.it, navarra@dipmat.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

C. Lavault and V. Ravelomanana are with the Laboratoire d'Informatique de Paris-Nord, University of Paris 13, Villetaneuse, Paris, France, {Christian.Lavault,vlad}@lipn.univ-paris13.fr

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

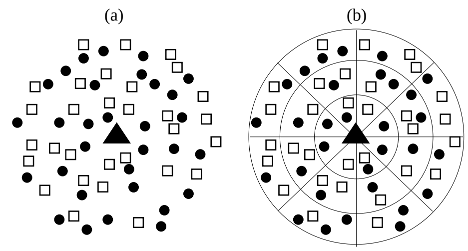

Fig. 1. (a) *An actor-centric heterogeneous sensor network, where the triangle in the middle represents the actor, circles represent free sensors, and squares are periodic sensors.* (b) *The sensor network where training has imposed a virtual coordinate system consisting of concentric coronas and equiangular sectors.*

As a fundamental prerequisite for self-organization, sensors need to acquire some form of location awareness, see [7], [10]. Almost all applications benefit from sensed data be supplemented with location information, but not all of them require the exact geographic position. Moreover, exact fine-grain location awareness usually assumes that the sensors are GPS-equipped. Therefore, massively deployed wireless sensors networks, which consist of very tiny sensors, can only be endowed with coarse-grain location awareness. The task of acquiring such a coarse-grain location, relative to a reference point, is referred to as *training*.

Recent papers have studied training protocols which impose a coordinate system by a single, more powerful device, referred to as *actor*, deployed in the wireless sensor network, see [1], [2], [8]. In support of its mission, the actor node is provided with a steady power supply, and a special radio interface for long distance communications. In particular, the actor has a directional antenna and can modulate the power transmitted so that its transmissions cover areas with different radii and different angles. When the actor transmits, all the awake sensors belonging to the area covered by the current transmission passively receive the actor message. The potential of such an actor to train the sensors has been explored in [1], [2], [3], [9], [11], [12] where training protocols are presented which divide the sensor network area, which consists in a disk around the actor, into equiangular sectors and concentric coronas, centered at the actor.

The training protocols reported thus far work on *homogeneous* sensor networks, that is, networks whose sensors are all *identical* in terms of computing and communication capabilities as well as of energy budget. In contrast, we study actor-driven training protocols working in heterogeneous sensor networks. The heterogeneous wireless sensor networks assume sensors with different capabilities. Heterogeneity can be introduced to increase the dynamic nature and the adaptation capacity of the network (i.e., sensors that in similar conditions behave differently can be adopted to differentiate the energy consumption and eventually prolong the overall network lifetime) or can result from on-the-fly modifications of existing networks (i.e., different sensors may be used in different re-deployments). The heterogeneous wireless networks considered hereafter consist of a single actor, and tiny massively deployed sensors which can differ in the way they alternate between *sleep* and *awake* periods. When a sensor is awake, its CPU is active, along with its timer, and its radio is on. Instead, when a sensor is sleeping, its CPU is not active, its radio is off, and only its timer is on. Since the sensors rely on integrated, small-scale, non-rechargeable batteries and since a sensor drains much less energy in sleep mode than in active mode [4], in order to save energy, the sensors should spend most of their time in sleep mode, waking up for brief time periods only.

In this work, two types of sensors are considered: the *free* sensors, which alternate between sleep and awake periods whose frequency and length depend on the executed protocol, and the *periodic* sensors, which alternate between sleep and awake periods according to a predefined plan that cannot be altered by the protocol. Such devices can model the general behavior of sensor nodes with harvesting capability which collect energy during the sleep periods and perform their duties during the awake periods. In Figure 1, free and periodic sensors are depicted with circles and squares, respectively.

The main contribution of this work is to propose a new actor-driven corona training protocol for heterogeneous wireless sensor networks. The behavior of the actor is based on linear strength decrease transmissions alternating with full strength transmissions. On the other hand, the sensors perform a binary search among the actor transmissions to locate their correct corona. Although the two types of sensors are driven by the same actor protocol, they locally act in a different way. The sensors are anonymous and indistinguishable to the actor. Each sensor starts the training task when it wakes up for the first time, without any initial explicit synchronization. It is assumed that, during the training task, both sensors and actor measure the time in slots, which are equal in both lengths and phase. However, every time a sensor receives a transmission from the actor, it can re-phase its own slot. This makes the protocol resilient to sensor clock drift without using any standard clock-synchronization protocol.

The remainder of this work is organized as follows. Section II gives a brief survey on related training protocols. Section III first discusses the wireless sensor and actor network model and introduces the task of training. In the same section, the actor and the sensor behavior of the proposed protocol is described. Section IV exhibits the worst-case performance analysis of the protocol, in terms of the number of sleep/awake transitions per sensor and thus in terms of energy consumed. Section V presents an experimental evaluation of the performance, tested on randomly generated instances, confirming the analytical results, and showing a much better behavior in the average case. The performance is then compared with that of all the previous training algorithms known for the periodic sensors, showing that the new protocol requires fewer sleep/awake transitions, and hence consumes much less energy per sensor. Finally, Section VI offers concluding remarks.

II. Related Works

In this section, previously known protocols for training either only periodic or only free sensors are summarized. The *Flat* corona training protocol and its variants, *Flat+* and *TwoLevel*, proposed in [1], [2], [12], deal with a homogeneous network of periodic sensors. They are called *asynchronous* protocols because each periodic sensor learns the identity of the corona to which it belongs, regardless of the moment when it wakes up for the first time. On the other hand, the two protocols proposed in [3] deal with a homogeneous network of free sensors, and are fully synchronous.

In the Flat protocol, immediately after deployment, the actor cyclically repeats a transmission cycle which involves k broadcasts at successively decreasing transmission ranges, where k is the number of coronas. Each broadcast lasts for a slot and transmits a beacon equal to the identity of the outmost corona reached. On the other side, 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 behavior is repeated until the sensor learns the identity of the corona to which belongs. Each sensor, during the training task, 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. In each time slot when the sensor is awake, the entry of R corresponding to the beacon transmitted by the actor is set either to 0 when the sensor hears nothing or to 1 when the the sensor hears the transmission. A sensor which belongs to corona c continues until it verifies the *training condition*, that is until there are two consecutive entries of R, say $c-1$ and c such that $R_{c-1}=0$ and $R_c=1$.

The Flat$+$ 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 protocol, the sensor now fills entries of R relative to beacons not yet transmitted during its awake periods. Therefore, it can look ahead and skip its next awake period if the corresponding entries of R have already been filled. However, as proved in [1], [2], its worst case performance remains the same as Flat.

A further improvement, called the *Two-Level* protocol, follows a nesting approach in which the k coronas are viewed as k_1 macrocoronas of k_2 adjacent macrocoronas each. Precisely, each sensor is first trained to learn the macrocorona it belongs to and then to learn its microcorona inside its macrocorona. Although Two-Level is the most efficient protocol known so far, it cannot reduce the number of sensor sleep/awake transitions below the square root of the number of transitions needed by the Flat protocol [1]. However, the actor behaviour of Two-Level is designed ad hoc for periodic sensors and cannot handle the free ones.

In contrast, the two protocols presented in [3] assume that all the sensors are free and synchronized to the master clock running at the actor. Such two protocols can be thought as visits of complete binary/d-ary trees, whose leaves represent coronas, whose node preorder/BFS numbers are related to the time slots, and whose node inorder/BFS numbers are related to the actor transmission ranges, respectively. Exploiting the fully synchronized model and performing a distributed phase where the sensors that have already known their corona inform those in their neighborhood, such protocols require a logarithmic number (in the number of coronas) of sensor sleep/awake transitions and achieve an optimal square root time (also in the number of coronas) for terminating the training task. However, the need of a strong synchronization between the actor and the sensors makes this protocol difficult to be extended to train periodic sensors.

This paper presents an asynchronous protocol which not only improves over all the previously presented asynchronous protocols, by reducing the number of sleep/awake transitions to a logarithmic number thus matching the fastest synchronous protocol presented in [3], but also can simultaneously train both free and periodic sensors.

III. The Binary Training Protocol

In this section, the network model is described and the details of the corona training protocol are presented, where each individual sensor has to learn the identity of the corona to which it belongs regardless of its type and of the moment when it wakes up for the first time.

A. The Network Model

A heterogeneous wireless sensor network is assumed to consist of a single fixed actor, provided with a steady power supply and a special radio interface for long distance communications, as well as a set of heterogeneous sensors, massively and randomly deployed in the actor broadcast range as illustrated in Figure 1(a).

Time is ruled into slots. The sensors and the actor rely on equally long, in phase slots. If the slot measured at a sensor drifts from that at the actor, the sensor can easily rephase its slot every time it wakes up, as it will be shown in the protocol. The sensors operate subject to the following fundamental constraints:

- Sensors are *anonymous* – to assume the simplest sensor model, sensors do not need individually unique IDs;
- Each sensor has a modest non-renewable energy budget;
- Each sensor has no global information about the network topology, but can receive transmissions from the actor;
- 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.

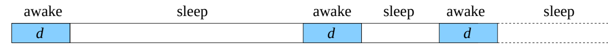

Fig. 2. *The irregular behaviour of a free sensor which alternates between awake periods of fixed length d and sleep intervals of arbitrary lengths.*

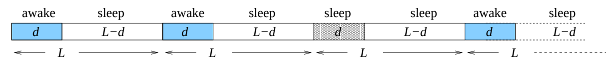

Fig. 3. *The sleep-awake cycle of a periodic sensor. The darkest d slots represent a time interval in which the sensor was scheduled to be awake but it decided to maintain its sleep mode.*

Two types of sensors can be distinguished according to their behavior:

- The *free* sensors alternate, according to internal computations, between sleep periods, whose lengths depend on the executed protocol and can assume arbitrary values, and awake periods of fixed length d.
- The *periodic* sensors alternate between sleep and awake periods, both of fixed length. The sensor sleep-awake cycle has a total length of L time slots, out of which the sensor is in awake mode for $d \leq L$ slots, periodic sensors can also sleep for their entire cycle, skipping awake periods, as depicted in Figure 3.

It is worth noting that the protocol to be discussed can simultaneously handle sensors each having its own sleep and awake parameters. Namely, the network may consist of free sensors each having a different awake period length d, and of periodic sensors each with distinct awake and sleep periods of length d and $L - d$, respectively. For the sake of simplicity, in the rest of this paper, it is assumed that the free and periodic sensors share the same awake period length d, that all the periodic sensors share the same sleep period length $L - d$, and that both d and L are even.

As a result of corona training, the deployment area is covered by k coronas $C_0, C_1, \ldots, C_{k-1}$ determined by k concentric circles, centered at the actor, whose radii are $r_0 < r_1 < \cdots < r_{k-1}$, as shown in Figure 1(b).

B. The Actor Behavior

The pseudocode of the actor behaviour is given in Figure 4. The actor repeats a cycle of $2k$ time slots. At time slots $2i$ and $2i+1$, with $0 \leq i \leq k-1$, the actor broadcasts a *control-broadcast*, followed by a *data-broadcast*. Both broadcast the beacon $k - 1 - i$, the former with a full power level able to reach the sensors lying in all the coronas, and the latter with a power level able to reach only those sensors up to corona C_{k-1-i}, but not those beyond. The actor transmission cycle, shown in Figure 5, is repeated for a time τ sufficient to accomplish the training protocol. An evaluation of τ will be given in Theorems IV.6 and IV.9 (for free and periodic sensors, respectively).

The redundancy of information between a control-broadcast and the subsequent data-broadcast allows the sensors and the actor to perform a light synchronization at any time during the training task. One reason for performing data-broadcasts in descending order is that the outer coronas, which have more sensors than the inner ones, are reached first. Moreover, since for free sensors, as proved in Lemma IV.4, the inner

```
Procedure Actor (k);
  t := 0;
  repeat
  for i := 0 to k − 1 do
      transmit beacon k − 1 − i up to corona C_{k−1};
      transmit beacon k − 1 − i up to corona C_{k−1−i};
  t := t + 2k;
  until t ≤ τ
```

Fig. 4. *The Binary protocol for the actor.*

coronas complete their training earlier than the outer coronas, a subnetwork connected to the actor grows and could start operating before the whole training task terminates.

C. The Sensor Behavior

In order to describe the protocol for the sensors, it is crucial to point out that the sensors are aware of the actor behavior and of the number of coronas k. Nonetheless, the control-broadcast could be used to pass global information, like k or the order (i.e., decreasing, increasing) in which the data-broadcast transmissions are scheduled. Moreover, it is worthy to recall that, during the training task, an awake sensor belonging to a generic corona c always receives a control-broadcast of any beacon, it cannot receive a data-broadcast of any beacon smaller than c, and receives the data-broadcast of any beacon larger than or equal to c.

First the behavior of any sensor, independently of its type, is sketched. To figure out its corona, a sensor uses two $(\lfloor \log k \rfloor + 1)$-bit registers, named *min* and *max*. At any instant, the *min* (*max*) register keeps track of the largest (smallest) corona, heard so far via a control-broadcast (data-broadcast), smaller than (larger than or equal to) the corona to which the sensor belongs. From now on, the interval $[min+1, \ldots, max]$ is called the *corona identity range*, and its width $max - min$ is denoted by λ. From the above discussion, the following *training condition* is verified:

Lemma III.1. *A sensor which belongs to corona* c*, with* $c \geq 0$*, is trained when* $\max = c$ *and* $\min = c - 1$*, and hence* $\lambda = 1$*.* □

Immediately after the deployment, each sensor wakes up at random within the 0-th and the $2(k-1)$-th time slot and starts listening to the actor for d time slots, with $d \geq 2$. During the awake period, the sensor properly sets the min and max registers according to the actor transmissions that it can receive. Then, to find the corona to which it belongs, each sensor alternates sleep and awake periods. Precisely, after the first awake period, each sensor guesses to belong to corona $\lceil \frac{\min + \max}{2} \rceil$ and goes to sleep until the actor transmits such a corona. At its next awakening, if the sensor receives the data-broadcast relative to the corona it guessed, its corona identity range becomes $[C_{\min+1}, \ldots, C_{\lceil \frac{\min+\max}{2} \rceil}]$, whereas if the sensor does not receive it, the corona range becomes $[C_{\lceil \frac{\min+\max}{2} \rceil + 1}, \ldots, C_{\max}]$. Such a binary search continues until the range boundaries differ by one, and thus the sensor is trained.

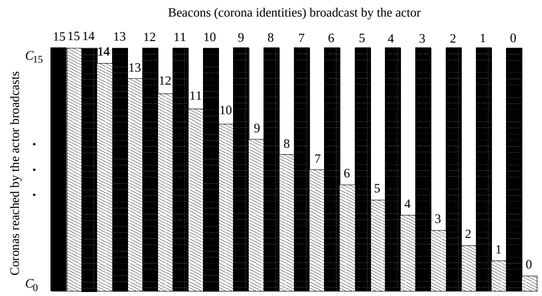

Fig. 5. *An actor transmission cycle of* $2k$ *time slots with* $k = 16$*. The actor alternates control-broadcasts at full power level (black) with data-broadcasts at decreasing power levels (gray), transmitting corona identities in decreasing order.*

```
    Procedure Binary-Training (k, d);
1     trained := false; ν := t := 0; min := −1; max := k − 1;
2     while ¬ trained do
3       for i := 0 to d − 1 do
4           if even(i) then
5               if received beacon c then
6                   first := c;
                else
7                   first := k;
            else
8               if ¬ received beacon c then
9                   if min ≤ first then
10                      min := first; update:=left;
11                      control:= t + i − 1;
                else
                    cases
12                      c = first:
13                          if max ≥ c then
14                              max := c; update:=right;
15                              control:= t + i − 1;
16                      first≠ k and c = (first−1) mod k:
17                          if max ≥ first then
18                              max := first; update:=right;
19                              control:= t + i;
20                      first = k:
21                          if min ≤ (c + 1) mod k then
22                              min := (c + 1) mod k;
23                              update:=left;
24                              control:= t + i;
25          t := t + d − 1;
26      if max − min = 1 then
27          mycorona := max;
28          trained := true;
        else
29          guess:= ⌈(min + max)/2⌉;
30          alarm-clock := control + Wait();
31          sleep until alarm-clock rings;
```

Fig. 6. *Training protocol for a generic sensor.*

The pseudocode for the sensor behavior is given in Figure 6. A sensor listens for an awake period of d consecutive time slots. Since the sensor is asynchronous, it keeps track of two slots, one even and one odd, to understand whether it woke up at a data-broadcast or a control-broadcast. During the even slots, it stores in variable *first* either the beacon received, if any, or k (lines 4–7). During the odd slots, if the sensor does not receive any beacon, it is sure that it woke up at a control-broadcast. Thus, the actor is now data-broadcasting the beacon *first* and the corona of the sensor must be larger than $first$. In variable *control*, the sensor remembers the local time when the control-broadcast was received (lines 8–11).

On the other hand, if during the odd slots the sensor receives beacon c, three cases arise depending on what happened in the previous slot, namely, a control-broadcast was received (lines 13–15), a data-broadcast was received (lines 17–19), or a data-broadcast was not received (lines 21–24). The first case is detected because the sensor hears the same beacon c

Function Wait: **integer**;
  if update $=$ right **then**
    Wait $:= 2\lfloor \frac{\max - \min}{2} \rfloor$;
  else
    Wait $:= 2\left(k - \lfloor \frac{\max - \min}{2} \rfloor\right)$;

Fig. 7. *The* Wait *procedure invoked for the free sensors.*

twice, which implies that the sensor belongs to a corona whose identity is smaller than or equal to c. The second case happens when the sensor hears two distinct beacons differing by $1 \bmod k$, yielding that the sensor belongs to a corona smaller than or equal to *first*. The third case occurs when the sensor hears only the beacon c during the second slot, with the consequence that the sensor belongs to a corona larger than $(c+1) \bmod k$. At the end of the awake period, the sensor tests the training condition (lines 26–28), and if it is not trained, by invoking the *Wait* procedure, the sensor intends to wake up again when the actor broadcasts the corona identity in the middle of the sensor corona identity range (lines 29–31). The time complexity of the *Binary-Training* protocol is $O(d)$ plus the time required for executing the *Wait* procedure.

So far, the behavior of the sensors during the Binary-Training process has been described independent of their type. Indeed, only the procedure *Wait*, which determines how long a sensor has to sleep in order to receive the *guess* corona, depends on the sensor type. Such a procedure mainly influences the total time each sensor employs to be trained, and thus the total time τ of the training task.

In the following, the *Wait* procedures, one for free and one for periodic sensors, are devised and analyzed.

1) The Free Sensor Behavior: This subsection deals with sensors that can freely choose their awakening time. So they set the alarm clock when, according to their local time, the actor transmits the *guess* corona identity.

The *Wait* procedure is outlined in Figure 7. The sensor sleeps for an interval which depends on the *guess* corona and the last modified boundary of the corona identity range.

Consider a sensor that finishes its current awake period and invokes the *Wait* procedure. If *update*=*right*, then *max* is the beacon transmitted via a control-broadcast at time slot *control* (see Figure 6). Since the *guess* corona is smaller than *max*, *guess* will be broadcast in the current actor cycle at time slot *control*$+2\lfloor \frac{\max - \min}{2} \rfloor$. Whereas, if *update*=*left*, then *min* has been transmitted by the actor at the beginning of the current awake period. Since *guess* can only be larger than *min*, *guess* will be transmitted during the next actor cycle, at slot *control*$+ 2(k - \lceil \frac{\max - \min}{2} \rceil)$. Clearly, the time complexity of the *Wait* procedure for the free sensors is $O(1)$.

Note that the sensor, setting the *control* variable, intends to wake up in the next period when the actor is transmitting the control-broadcast relative to the *guess* corona. However, the pseudo-code does not exploit this property. Indeed, the protocol works properly even if the sensor wakes up again when any data-broadcast or control-broadcast is transmitted. Moreover, since the sensor updates the *min* and *max* registers listening to the effective actor transmission, the sensor does not infer any information from its knowledge of the actor behavior, contrary to the previously known protocols [1], [2], [3]. For all these reasons, the new protocol is robust to possible clock driftings.

Function Wait: **integer**;
1 **if** update $=$ right **then**
2   firstcorona $:= |\max + \frac{d}{2}|_k$;
  else
3   firstcorona $:= \min$;
4 $\gamma := 1$;
5 **while** guess $\not\in \left[|\text{firstcorona} - \gamma\frac{L}{2} - \frac{d}{2} + 1|_k, |\text{firstcorona} - \gamma\frac{L}{2}|_k\right]$ **do**
6   $\gamma := \gamma + 1$;
7 Wait $:= \gamma L - d + 1$;

Fig. 8. *The* Wait *procedure invoked for the periodic sensors.*

2) The Periodic Sensor Behavior: In this subsection, the *Wait* procedure for the periodic sensors is devised. For the sake of the analysis, each sensor is assumed to wake up for the first time at a random instant $2s$, with $0 \leq s \leq k-1$. Recall that a sensor running this protocol always alternates d slots during which it is awake and $L-d$ slots in which it sleeps. In each of the d slots where the sensor is awake, it updates its position according to the sensed data. At each awakening, each sensor hears groups of $\frac{d}{2}$ consecutive corona identities, broadcast by the actor. Since two consecutive awake periods start L time slots apart, the corresponding first beacons transmitted by the actor are $|\frac{L}{2}|_k$ apart. Hence, a periodic sensor which does not skip any awake period hears the k coronas in a specific order which depends on the parameters d, L, and on the time slot s at which the sensor wakes up for the first time. On the top of such an order, the Binary-Training protocol imposes the binary search scheme on the corona identity range by means of the *Wait* procedure, which forces a sensor to skip awake periods until that in which *guess* is transmitted.

The *Wait* procedure is given in Figure 8. Consider a sensor that finishes its current awake period at slot t and invokes the *Wait* procedure. At first, the sensor recomputes in variable *firstcorona* the beacon which was transmitted by the actor at the beginning of its current awake period. Indeed, if *update*=*right*, then max has been updated at each time slot and *firstcorona* is $|\max + \frac{d}{2}|_k$. Whereas if *update*=*left*, then min has been updated only at the first time slot of the awake period, and *firstcorona* is exactly the corona identity stored in register *min* (lines 2-3). Note that in the first awake period, if two boundaries have been updated, register *update* must be equal to *right*. Thus, in a lookup process, the sensor checks during which subsequent awake period the *guess* corona will be transmitted (lines 5–6), and stores in γ the number of awake periods to be skipped plus one. Indeed, since the corona identities transmitted at the beginning of two consecutive awake periods differ by $|\frac{L}{2}|_k$, and $\frac{d}{2}$ beacons are transmitted in each awake period, the sensor knows which beacons it can receive in every awake period.

The time complexity of the Wait procedure, shown in Figure 8, is $O(\gamma d)$. However, such a complexity can be lowered by storing in each sensor a look-up table, as it will be shown at the end of Subsection IV-B.

IV. Correctness and Performance Analysis

In this section, the correctness and the performance of the Binary-Training protocol are discussed. The results proved in the next lemmas hold for both free and periodic sensors.

Lemma IV.1. *Each sensor requires at least* 2 *consecutive time slots to learn its relative position with respect to the beacon transmitted in the last data-broadcast.*

Proof: By contradiction, consider a sensor that listens to the actor for just one slot. If the sensor receives beacon c, it cannot distinguish whether it hears a control- or a data-broadcast. On the other hand, if the sensor does not receive any beacon, although it is aware that the actor transmits a data-broadcast, it cannot update the *min* register because it does not know the transmitted beacon. Therefore, in both cases the sensor cannot update its corona identity range. Consider now a sensor that has listened for two consecutive time slots. Since $i = 1$, the sensor executes the code in lines 8–23, and hence it sets either min or max learning its relative position with respect to the last data-broadcast beacon. □ ■

As a consequence of Lemma IV.1, it is necessary that the length d of the awake period of both free and periodic sensors be at least 2 to allow all the sensors to be trained (such a condition is also sufficient only for free sensors, as it will be shown later). Let us now concentrate on how the width λ of the corona identity range decreases for any sensor. Precisely, in the first awake period of a sensor, λ reduces as follows:

Lemma IV.2. *Consider a sensor belonging to corona* c *that wakes up at time slot* s, $0 \leq s \leq 2k-1$, *when the actor transmits beacon* K_s, *with* $0 \leq c, K_s \leq k-1$. *If the sensor is untrained at the end of the first awake period, the width* $\lambda = max - min$ *of its corona identity range is:*

$$\lambda = \begin{cases} \textit{min}\{k - K_s - 1, k - \frac{d}{2}\} & \textit{if } c > K_s \\ K_s - \frac{d}{2} + 1 & \textit{if } c \leq K_s \end{cases}$$

Proof: Consider the behavior of a sensor that at the end of its first awake period is still untrained. Assume that the sensor does not receive the data-broadcast transmitting beacon K_s, that is, $c > K_s$. If $K_s \geq \frac{d}{2}$, then the *min* boundary of its corona identity range is updated to K_s. Since the actor transmits at decreasing power levels, the next d transmissions will not update register *min*. Hence, the corona identity range becomes $[K_s+1, \ldots, k-1]$. Whereas, if $K_s < \frac{d}{2} - 1$, also the register *max* is updated because the sensor is awake while the actor transmits beacon $k-1$. However, overall $\frac{d}{2}$ coronas are excluded, leading to a corona range of width $k - \frac{d}{2}$. Assume now that the sensor receives the data-broadcast transmitting beacon K_s, that is, $c \leq K_s$. Then, the sensor updates the *max* boundary for $\frac{d}{2}$ times. Therefore, the new corona range becomes $[0, \ldots, K_s - \frac{d}{2}]$. Note that if $K_s < \frac{d}{2}$, the sensor will be trained. □ ■

The following two results hold for trainable sensors, that is, for those sensors that after a finite time have $\lambda = 1$.

Lemma IV.3. *In each awake period but the first, every trainable sensor, which belongs to corona* $c > 0$, *updates only one boundary of its corona identity range unless it becomes trained. Every sensor in corona* 0, *always updates only one boundary.*

Proof: By contradiction consider a sensor in corona $c > 0$ that updates both boundaries in the same awake period, but remains untrained. Let *min* and *max* be the values of the boundaries at the beginning of the awake period. During such an awake period, the sensor must have received the control-broadcast for a corona larger than *min* down to the data-broadcast for a corona smaller than *max* (passing through the control-broadcasts for coronas 0 and $k-1$). However, this takes more than d time slots since, already at the end of the first awake period, at least $(min+1) + (k - max) \geq \frac{d}{2}$ coronas are excluded by the corona identity range.

If a sensor belongs to corona 0, since whenever it wakes up it receives the actor transmission, it sets *max* in each awake period. When it receives beacon 0, it is trained because $max - min = 0 - (-1) = 1$. □ ■

As explained in Subsection III-C, in each awake period but the first, the width λ of the corona identity range is reduced by applying a binary search scheme on the interval $[min, \ldots, max]$ until $\lambda = 1$. This process requires a number of sleep/awake transitions, whose worst value is denoted by $\nu_{\max}$, bounded as follows:

Lemma IV.4. *A trainable sensor that belongs to corona* c *and wakes up for the first time at time slot* s, $0 \leq s \leq 2k-1$, *while the actor transmits beacon* K_s, *with* $0 \leq c, K_s \leq k-1$, *requires*

$$\nu_{\max} \leq \begin{cases} 1 + \lceil \log(\min\{k - K_s - 1, k - \frac{d}{2}\}) \rceil & \textit{if } c > K_s \\ 1 + \lceil \log(K_s - \frac{d}{2} + 1) \rceil & \textit{if } c \leq K_s \end{cases}$$

transitions to be trained.

Proof: After the first awake period, the corona identity range reduces by half at each awakening because the sensor learns its relative position with respect to the *guess* corona, which is in the middle of the corona identity range. Therefore, by Lemma IV.2, the result follows. □ ■

It is worth noting that a free sensor is always trainable provided that $d \geq 2$ because, being free to set its alarm-clock, it is guaranteed to hear the *guess* corona. In contrast, a periodic sensor is constrained in its awakenings and thus it is trainable only if some conditions on the parameters L, k and d are verified, as it will be proved in Subsection IV-B.

In order to analytically evaluate the performance of the *Binary-Training* protocol, in addition to ν_{max}, let ω_{max} be the worst overall awake time per sensor, and τ be the total time for training. Recalling that each awake period lasts for d time slots, one has $\omega_{max} = \nu_{max} d$. Note that τ measures the time required to terminate the whole training task for the actor, whereas each sensor counts in t its *local* training time, that is, how many slots elapse from the sensor first wake up until the end of the awake period in which it is trained. Hence, a sensor which is trained at local time t is trained at time $t + s$ for the actor, if s is the random time slot when it wakes up the first time. Therefore, τ cannot be larger than $t_{max} + 2k - 1$, where t_{max} is the worst training time among the training times of all the sensors. The analysis of the total time required by Binary-Training depends on the *Wait* procedure, which determines how long a sensor has to sleep before receiving the *guess* corona, and hence it is different for free and periodic sensors.

A. Free Sensors

In order to bound from above the total time τ for the training task, the following result is useful:

Lemma IV.5. *The training task for a free sensor that belongs to corona c cannot last more than $\tau_c = 2k(1 + \lceil \log_2 c \rceil)$ time slots. Therefore, $\tau \leq 2k(1 + \lceil \log_2 k \rceil)$.*

Proof: By applying the binary search scheme to the corona identity range, a sensor that belongs to corona c must exclude the coronas $0, 1, \dots, c-1$ from its corona identity range by updating the register *min*. This can be done in at most $\lceil \log_2 c \rceil$ times. Since the sensor waits at most $2k$ slots between two consecutive updates of *min*, the result follows. □ ■

A consequence of the above lemma is that the inner coronas finish the training task earlier than the outer coronas. In this way, the wireless sensor network is raised up from the center to the periphery. Hence, the performance of the Binary-Training protocol for free sensors can be summarized as follows:

Theorem IV.6. *All the free sensors are trainable if $d \geq 2$ and each free sensor requires to be trained $\nu_{\max} \leq 1 + \lceil \log_2 k \rceil$, $\omega_{\max} = d\nu_{\max}$, and $\tau \leq 2k\nu_{\max}$.*

Proof: The proof follows from Lemmas IV.1, IV.4, and IV.5. □ ■

B. Periodic Sensors

To analyze the performance of the Binary-Training protocol for periodic sensors, some properties on which beacons are received by the sensor, and in which order, are discussed. Denote with (a, b) the greatest common divisor between a and b (see [6]), and let $L' = \frac{L}{2}$, $g = (L', k)$, $d' = \frac{d}{2}$, $\hat{L}' = \frac{L'}{(L',k)}$, and $\hat{k} = \frac{k}{(L',k)}$. In order to derive the necessary and sufficient condition to train all the periodic sensors, the following observation is useful.

Lemma IV.7. *For fixed L, d, and k, assume that, during the first two slots, when the sensor wakes up for the first time, the actor has transmitted the data-broadcast K_s, with $0 \leq K_s \leq k-1$. Then the data-broadcast transmitted in the first two slots of the i-th sensor awake period is $|K_s - iL'|_k = \left| K_s - (L', k)|i\hat{L}'|_{\hat{k}} \right|_k$, assuming that the sensor does not skip any awake period. Overall only $\hat{k}$ different data-broadcasts can be transmitted by the actor in the first two slots of every sensor awake period, independent of how many awake periods the sensor performs. Such $\hat{k}$ data-broadcasts differ each other by a multiple of (L', k).*

Proof: Consider a sensor for which, during its first awake period, the data-broadcast K_s has been the first one transmitted by the actor and which does not skip any awake period. The i-th awake period, $i \geq 0$, of such a sensor starts iL time slots later while the actor is data-broadcasting, during the first two slots of the sensor awake period, $|K_s - iL'|_k = |K_s - |iL'|_k|_k$. Observe that L' and k can be rewritten as $L' = g\hat{L}'$ and $k = g\hat{k}$. Since $|iL'|_k = g|i\hat{L}'|_{\hat{k}}$ (see [6]), $|iL'|_k$ is a multiple of g and generates only the $\hat{k}$ multiples of g in $[0, \dots, k-1]$ while i varies in any interval of at least $\hat{k}$ consecutive integer values. Therefore, $|K_s - |iL'|_k|_k = \left| K_s - g|i\hat{L}'|_{\hat{k}} \right|_k$. Moreover, in any two awake periods, say the i-th and the j-th ones, such that $i > j$ and $i - j < \hat{k}$, the two first data-broadcasts transmitted are distinct and differ by a multiple of g. Whereas, the same first data-broadcasts is transmitted in any two awake periods i and j such that $i \equiv j \bmod \hat{k}$. □ ■

	0	1	2	3	4	5	6
0	0	7	6	5	4	3	2
1	2	1	0	7	6	5	4
2	4	3	2	1	0	7	6
3	6	5	4	3	2	1	0

Fig. 9. Table A showing the coronas broadcast by the actor during the awake periods of a sensor, assuming it does not skip any awake period and that it woke up for the first time while the actor was transmitting $K_s = 0$.

For example, assume $L = 28$, $k = 8$, and $d = 14$, and consider a sensor that wakes up for the first time while the actor broadcasts $K_s = 0$. In Figure 9, a table A is depicted which shows in row i the coronas heard at the i-th awake period. According to Lemma IV.7, A has $\hat{k} = \frac{8}{2} = 4$ rows and $d' = 7$ columns. For instance, column 0 shows the $\hat{k}$ different data-broadcasts $\{0, 2, 4, 6\}$ which can be transmitted in the first two slots of every sensor awake period and which differ by $g = (14, 8) = 2$. Instead, row 1 shows the 7 coronas broadcast during its second awake period, assuming that the sensor does not skip it. Observe that the first corona transmitted in this second awake period is $|K_s - iL'|_k = |0 - 1 \cdot 14|_8 = 2$. If the sensor does not skip any awake period, it wakes up in the next two awake periods while the actor transmits 4 and 6, respectively, as depicted in column 0. This behaviour is periodic and in any subsequent awake period the sensor will wake up while the actor broadcasts one corona among $\{0, 2, 4, 6\}$.

As a consequence of Lemma IV.7, a sensor can hear, regardless of how long the training task lasts, $\hat{k}$ distinct sequences each of d' consecutive decreasing coronas. If $d' < (L', k)$, the sensor receives $\hat{k}$ non-overlapping sequences of coronas, and hence only $\hat{k}d' < k$ coronas. If $d' \geq (L', k)$, the sensor hears at least once each of the k coronas.

Lemma IV.8. *The training condition is satisfied for all the periodic sensors if and only if $d' \geq (L', k)$.*

Proof: By Lemma IV.7, regardless of how long the training task lasts, a sensor can learn its relative position only respect to $\min\{k, d'\hat{k}\}$ different coronas even if it does not skip any awake period. Therefore, if $d' \geq (L', k)$, since $\min\{k, d'\hat{k}\} = k$, for any *guess* corona of the Binary-Training protocol there is at least one of the subsequent $\hat{k}$ consecutive awake periods in which the sensor can hear *guess*. Whereas, if $d' < (L', k)$, since $\min\{k, d'\hat{k}\} = d'\hat{k} < k$. there are coronas which can never be heard by the sensor irrespective of the training task duration. If one of such coronas is a *guess* corona for a sensor, the protocol cannot terminate for such a sensor, which thus remains untrained. □ ■

Therefore, the performance of the Binary-Training protocol for periodic sensors is given by the following result.

Theorem IV.9. *For fixed L, d, and k, if $d' < (L', k)$ then there are sensors which cannot be trained by the Binary-Training protocol; otherwise all the periodic sensors require to be trained $\nu_{\max} \leq 1 + \lceil \log_2 k \rceil$, $\omega_{\max} = d\nu_{\max}$, and $\tau \leq \hat{k} L \nu_{\max}$.*

Proof: The results for $\nu_{\max}$ and $\omega_{\max}$ follow from Lemma IV.4. With regard to τ, since the cycles of the actor and of the sensors last $2k$ and L slots, respectively, then the actor and the sensors are simultaneously at the beginning of their cycle every $l.c.m.\{2k, L\} = \frac{2kL}{(L,2k)} = \hat{k}L$ slots. In other words, the cycle of the actor-sensor system, i.e., the minimum time after which both the actor and a sensor are again in the initial condition, is of $\hat{k}L$ slots. Since to hear each *guess* corona a sensor has to wait at most a cycle of the actor-sensor system, and since at most $\nu_{\max}$ guesses are performed, the protocol takes $\tau \leq \hat{k} L \nu_{\max}$ time slots. □ ■

However, taking into account the particular values that d can assume, better bounds on the performance parameters can be derived.

Theorem IV.10. *For fixed L, d, and k, one has:*

1) *if $(L', k) \leq d' < |L'|_k$, then $\nu_{\max} \leq 1 + \lceil \log \frac{k}{d'} \rceil$, $\omega_{\max} = d\nu_{\max}$, and $\tau \leq \frac{k}{d'} L \nu_{\max}$;*
2) *if $|L'|_k \leq d' < k$, then $\nu_{\max} \leq 1 + \lceil \log \frac{k}{d'} \rceil$, $\omega_{\max} = d\nu_{\max}$, and $\tau \leq \lceil \frac{k}{d'} \rceil L \nu_{\max}$;*
3) *if $d' = k$, then $\nu_{\max} = 1$ and $\omega_{\max} = \tau = d$.*

Proof: The result trivially follows when $d' = (L', K)$ because, by Lemma IV.7, the k coronas are partitioned into $\frac{k}{d'}$ non-overlapping intervals over which a binary search is performed to find where *guess* is transmitted. Hence, the binary search takes $\nu_{\max} = 1 + \lceil \log \frac{k}{d'} \rceil$ guesses. Since each interval lasts $d = 2d'$ slots and since a sensor waits at most $\frac{k}{d'} L$ slots to hear each *guess* corona, the results for $\omega_{\max}$ and τ follow.

When $d' = |L'|_k$, if the sensor is awake for two consecutive awake periods, that is, for two awake periods starting at time slot t and $t+L$, it would hear $c-d'+1$ as the last corona of the fist period and $c-d'$ as the first corona of the second period, if c is the corona heard at time t. Thus, the k coronas are covered by $\lceil \frac{k}{d'} \rceil$ intervals (out of which $\lfloor \frac{k}{d'} \rfloor$ are non-overlapping) and a binary search is performed on such intervals to find where *guess* is transmitted. Since each interval lasts $d = 2d'$ slots and since a sensor waits at most $\lceil \frac{k}{d'} \rceil L$ slots to hear the *guess* corona, the bounds for $\omega_{\max}$ and τ hold.

When $d' = k$, the k coronas are covered in a single interval, and each sensor is trained in the first awake period. Thus, the bounds are trivially derived.

Observe that when $(L', k) < d' < |L'|_k$ or $|L'|_k < d' < k$, the number of intervals which cover the k coronas cannot be greater than that in the case of $d' = (L', k)$ and $d' = |L'|_k$, respectively. Hence, the proof follows. □ ■

With regard to the time complexity of the *Wait* procedure (Fig. 8), one can use a table T_{K_s}, where K_s is defined in Lemma IV.7, to faster compute γ. T_{K_s} consists of k rows and $\lceil \frac{d'}{g} \rceil$ columns. Given h and j, with $0 \leq h \leq k-1$ and $0 \leq j \leq \lceil \frac{d'}{g} \rceil - 1$, a generic entry $T_{K_s}(h, j)$ contains the awake period in which the sensor will hear the corona identity h in a time slot included between the time slots jg and $(j+1)g - 1$, where jg and $(j+1)g$ are the coronas where two awake periods start. The value $T_{K_s}(h, j)$ verifies $0 \leq T_{K_s}(h, j) \leq \hat{k} - 1$ and it is intended as a relative position within the system actor-sensor cycle. In practice, row h of T_{K_s} contains all the awake periods in which the sensor can hear corona h during the system actor-sensor cycle if the sensor does not skip any awake period. It is worth noting that the same corona can be heard by a sensor in more than one awake period (unless $d' = g$, in which case there is only a single column in T_{K_s}). Indeed, since each awake period includes d' consecutive coronas and since distinct awake periods start with coronas which are multiples of g, corona h is heard by a sensor in at most $\lceil \frac{d'}{g} \rceil$ awake periods, namely, for all those overlapping periods which include h.

Referring to the example in Figure 9, Figure 10 shows the content of T_0 for the same parameters, namely, $L = 28$, $k = 8$, and $d = 14$. For instance, row 5 of T_0 contains $T_0(5, 0) = 3$, $T_0(5, 1) = 0$, $T_0(5, 2) = 1$, and $T_0(5, 3) = \infty$, because corona 5 is transmitted during the 3-rd awake period in one of the slots 0 and 1, during the 0-th awake period in one slot between 2 and 3, in the 1-th awake period in one slot between 4 and 5, while it is never transmitted in slot 6, as one can check in Figure 9.

To better understand how to build table T_{K_s}, one could imagine to first construct a table A_{K_s} by setting $A_{K_s}(u, v) = |K_s - uL' - v|_k$. Since A_{K_s} contains the coronas heard in each awake period by a sensor that wakes up for the first time in corona K_s, one can derive the entries of T_{K_s} performing a kind of inverse computation. Precisely, if $A_{K_s}(u, v) = h$, with $0 \leq u \leq \hat{k} - 1$ and $0 \leq v \leq d' - 1$, then $T_{K_s}(h, \lfloor \frac{v}{g} \rfloor)$ is set to u. The unfilled entries in the last column of T_{K_s}, if any, are set to ∞. Clearly, this requires $O(\frac{kd'}{g})$ time and $O(\frac{kd'}{g} \log k)$ space for each sensor.

The above computation can be performed by each sensor at the beginning of the protocol, as soon as it knows its own K_s. Otherwise, such a computation can be done in a preprocessing phase, that is, before the sensor deployment, for a fixed value of K_s, like $K_s = 0$. In the latter case, when $K_s \neq 0$, each entry of T_{K_s} can be derived by the sensor from the precomputed table T_0 as: $T_{K_s}(h, j) = T_0(|h - K_s|_k, j)$. In other words, T_{K_s} corresponds to a row cyclic shift of T_0.

Finally, the number γ required in the *Wait* procedure of Figure 8 is obtained in $O(\frac{d'}{g})$ time by computing

$$\gamma = \min_{0 \leq j \leq \lceil \frac{d'}{g} \rceil - 1} \{\gamma_j : \gamma_j > 0\}$$

where

$$\gamma_j = |T_{K_s}(guess, j) - T_{K_s}(firstcorona, 0)|_{\hat{k}}\,.$$

In fact, one computes the minimum number γ_j of the awake periods between each occurrence of the *guess* corona, $T_{K_s}(guess, j)$, in the system actor-sensor cycle and the current awake period, given by $T_{K_s}(firstcorona, 0)$.

	0	1	2	3
0	0	1	2	3
1	1	2	3	∞
2	1	2	3	0
3	2	3	0	∞
4	2	3	0	1
5	3	0	1	∞
6	3	0	1	2
7	0	1	2	∞

Fig. 10. The table T_0 indicating the awake periods in which each corona is heard by a periodic sensor when $L = 28$, $k = 8$, and $d = 14$.

TABLE I
ESTIMATE OF SENSOR POWER CONSUMPTION IN DIFFERENT OPERATIONAL MODES AT 2.5 VOLT.

Sensor Mode	**Current Draw**	**Power Consume**
CPU inactive, timer on, radio off	6 μA	0.015 mW
CPU switch on, radio startup	3 mA	< 30 mW
CPU switch off, radio shutdown	3 mA	< 30 mW
CPU active, radio listening or RX	12 mA	32 mW

For example, consider a sensor with $K_s = 0$, which has $guess = 5$ and $firstcorona = 2$. Since $T_0(2,0) = 1$, one has $\gamma_0 = |T_0(5,0) - 1|_4 = 2$, $\gamma_1 = |T_0(5,1) - 1|_4 = 3$, $\gamma_2 = |T_0(5,2) - 1|_4 = 0$, and $\gamma_3 = |\infty - 1|_4 = \infty$. Hence, $\gamma = \min\{2, 3, \infty\} = 2$, and the sensor has to wait $2L - d + 1 = 56 - 14 + 1 = 43$ slots.

C. Energy Consumption

In this subsection, the energy drained by the Binary-Training protocol is evaluated under a realistic estimate of the power consumed by the sensors in their different operative modes.

During the training task, when a sensor is awake, its CPU is active and its radio is listening or receiving. Instead, when a sensor is sleeping, its CPU is not active, its timer is on, and its radio is off. Let e_{awake} and e_{sleep} be the energy consumed during a time slot by a sensor when it is listening/receiving or sleeping, respectively. Since the radio startup and shutdown require a non negligible overhead, let e_{trans} denote the energy consumed for a sleep/awake transition followed by an awake/sleep transition. Thus, denoted with ν and ω, respectively, the number of wake/sleep transitions and the overall awake time, the total energy E depleted by a sensor is:

$$E = \nu e_{\text{trans}} + \omega e_{\text{awake}} + (\tau - \omega) e_{\text{sleep}} \quad (1)$$

An upper bound on the energy drained by the training protocol for a free sensor is obtained from Equation 1 by substituting the worst case bounds for ν, ω, and τ given in Theorem IV.6, thus having:

$$E < (1 + \lceil \log k \rceil)\,(e_{\text{trans}} + d e_{\text{awake}} + 2k e_{\text{sleep}})$$

Similarly, the energy spent by the protocol for periodic sensors is derived from Equation 1 by using the upper bounds provided in Theorem IV.10, observing that $d' \geq (L', k)$:

$$E < \left(1 + \log \left\lceil \frac{k}{(\frac{L}{2}, k)} \right\rceil \right) \left(e_{\text{trans}} + d e_{\text{awake}} + \frac{kL}{(\frac{L}{2}, k)} e_{\text{sleep}} \right)$$

In order to evaluate the energy drained in a realistic setting, Table I reports the power consumed by a sensor in different operational modes. The data refer to the TinyNode 584, produced by Shockfish S.A., and are the customary values for the smallest sensors one can buy [4]. The sensors have as a power source two customary 1.2 Volt batteries, with a capacity of 1900 mAh each, and hence they have an energy supply of 4.56 Joule. As one can check in the table, listening is nearly as expensive as receiving. The radio startup and shutdown require a power consumption, which cannot be higher than that in the active mode, and they take a non negligible amount of time (about 1 ms each). The above constraint influences the behavior of the protocol because it gives a lower bound on the the length of the sensor sleep period, which must be sufficient to allow both radio startup and shutdown, and thus cannot be shorter than 2 ms. Hence, a time slot of 2 ms is enough. Note that such a slot duration is enough to accommodate within it the $O(\frac{d'}{g})$ computation time required in the worst case by a periodic sensor. In summary, from the data of Table I, one has that $e_{\text{trans}} = 30*1+30*1 = 60$ μJ, $e_{\text{sleep}} = 0.015*2 = 0.030$ μJ, and $e_{\text{awake}} = 32 * 2 = 64\ \mu J$.

It is easy to see that since the actual value of e_{sleep} is negligible with respect to e_{trans} and e_{awake}, which in turn are comparable, the periodic sensors, which require a smaller $\nu_{\max}$, consume slightly less energy than the free ones, which in turn are trained faster.

V. EXPERIMENTAL TESTS

In this section, the worst and average performance of the Binary-Training protocol are experimentally tested and compared with the asynchronous corona training protocols previously presented in [1], [2]. Since in the heterogeneous networks, the free sensors do not influence the performance of the periodic ones, and vice versa, the Binary-Training protocol has been tested training either only free or only periodic sensors. In this way, the comparison with the previous protocols, which deal only with homogeneous networks, is more evident. In particular, in this section, the protocol for free or for periodic sensors is called *BinFree* and *BinPeriodic*, respectively. In the simulation, there are $N = 10000$ sensors uniformly and randomly distributed within a circle of radius $\rho = k$, centered at the actor and inscribed in a square.

Consider first some experiments comparing the performance of the BinFree protocol versus the BinPeriodic one. In the simulations reported in Figures 11-15, the number k of coronas is fixed to 64, the length L of the sensor sleep-awake cycle is 216. Since the BinPeriodic protocol trains all the sensors only if $\frac{d}{2} \geq (\frac{L}{2}, k) = (108, 64) = 4$, the sensor awake period d varies between $2(\frac{L}{2}, k) = 8$ and $2k = 128$ with a step of 8. The results are averaged over 3 independent experiments.

Figure 11 shows the number of transitions for the different values of d. According to Theorems IV.4 and IV.9, when $d = 2(L', k) = 8$, BinFree and BinPeriodic have $\nu_{\max} = 1 + \lceil \log(k - \frac{d}{2}) \rceil = 7$ and $\nu_{\max} = 1 + \lceil \log(\frac{k}{\frac{d}{2}}) \rceil = 5$, respectively. Similarly, when $d = 2|L'|_k = 88$, BinFree and BinPeriodic require $\nu_{\max} = 6$ and $\nu_{\max} = 2$, respectively.

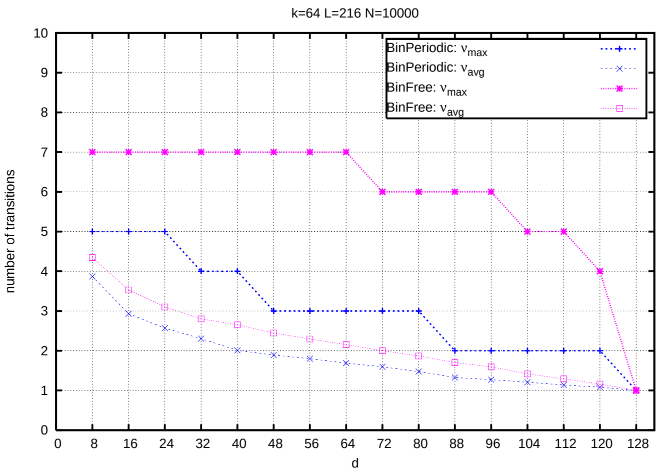

Fig. 11. *Number of transitions when $k = 64$, $L = 216$, and $8 \leq d \leq 128$.*

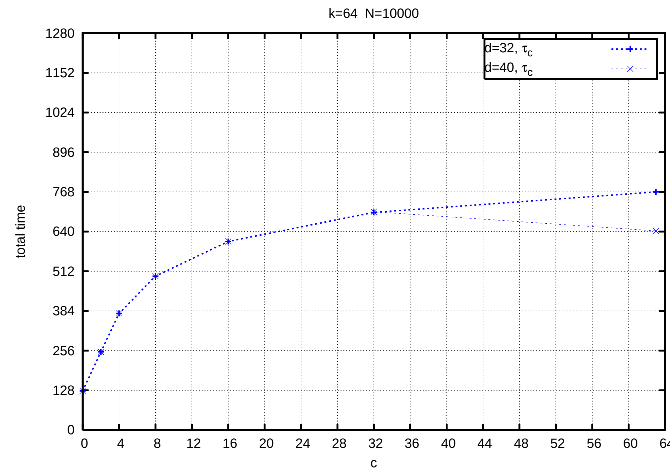

Fig. 13. *Total time slots required by the BinFree protocol to train all the sensors in corona $c = 2^i$, with $0 \leq i \leq 6$, when $k = 64$.*

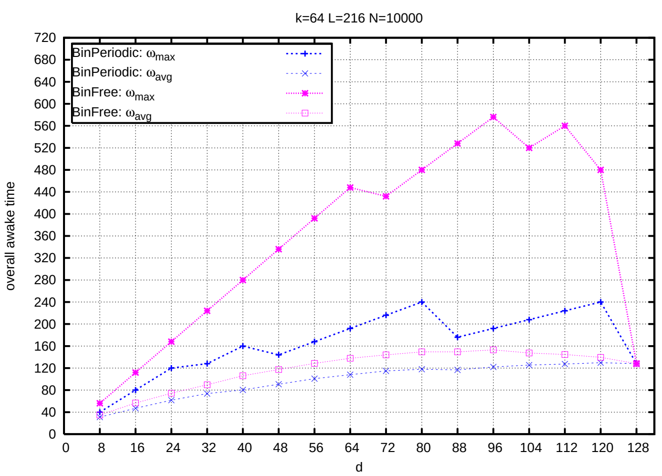

Fig. 12. *Overall sensor awake time slots when $k = 64$, $L = 216$, and $8 \leq d \leq 128$.*

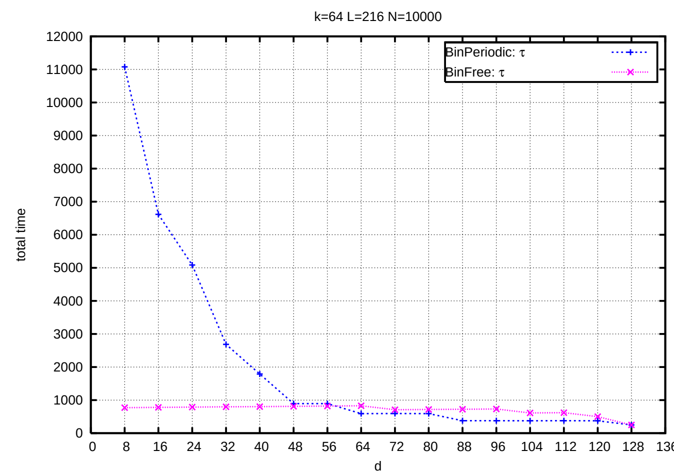

Fig. 14. *Total time slots when $k = 64$, $L = 216$, and $8 \leq d \leq 128$.*

Clearly, increasing d, the gain of BinPeriodic over BinFree increases. With regard to the average performance, although one notes that $\nu_{\rm avg}$ considerably improves over $\nu_{\rm max}$ for both protocols, the improvement is higher for BinFree.

Figure 12 presents $\omega_{\rm max} = \nu_{\rm max} d$ and $\omega_{\rm avg} = \nu_{\rm avg} d$, which measure, respectively, the worst and average overall awake time spent by each sensor to be trained. Clearly, BinFree exhibits awake times longer than those of BinPeriodic since it requires a larger number of transitions. Although the number of transitions decreases as d increases, Figure 12 illustrates that the average overall awake time is almost constantly slightly increasing for both protocols, except when d approaches $2k$, when all protocols take $\omega = 2k$. It is worthy to note that BinFree can train all the sensors even when $d = 2$, and in that case it achieves the absolute minimum for $\omega_{\rm max} = 2\nu_{\rm max} = 14$.

Figure 13 exhibits the total time τ required to accomplish the BinFree protocol for all the sensors in corona $c = 2^i$, with $0 \leq i \leq 6$, when $k = 64$, and either $d = 32$ or $d = 40$. The graphic confirms the results for the total time τ_c given in Lemma IV.5, that is $\tau \leq 2k(1 + \lceil \log_2 c \rceil)$. Figure 14 shows the total time τ required by the two protocols to train all the sensors in the network. BinPeriodic requires a total time extremely larger than that of BinFree when $d = 2(\frac{L}{2}, k) = 8$. In fact, for such a value of d, to receive the guessed corona, a free sensor has to wait at most $2k$ slots for each transition, whereas a periodic sensor has to wait at most $\frac{kL}{(L',k)}$ slots, that is a cycle of the actor-sensor system. The total time of the BinPeriodic protocol neatly decreases when d increases until it becomes comparable with that of BinFree for $d \geq \frac{k}{2}$. Indeed, when d is sufficiently large the coronas transmitted in different awake periods overlap. Hence, the same corona identity can be received by the periodic sensor during several awake periods of the same actor-sensor cycle, and in general, the sensor waits much less than $\frac{kL}{(L',k)}$ slots to receive the guessed corona. Note that the total time also decreases because, when d increases, the number of transitions required to train a sensor decreases.

Figure 15 shows the energy consumed by a sensor in the worst and average cases, denoted by $E_{\rm max}$ and $E_{\rm avg}$, respectively, for both the BinPeriodic and BinFree protocols, where the time slot length is set to 2 ms. It is worth noting that the graphic of the energy has the same profile as that of the overall awake time. In fact, since the actual value of $e_{\rm sleep}$ is negligible with respect to $e_{\rm trans}$ and $e_{\rm awake}$, which in turn are comparable, the energy grows proportionally to $\nu(d+1)$. Therefore, although BinPeriodic has a higher τ than BinFree when $8 \leq d \leq 48$, the former always consumes less energy

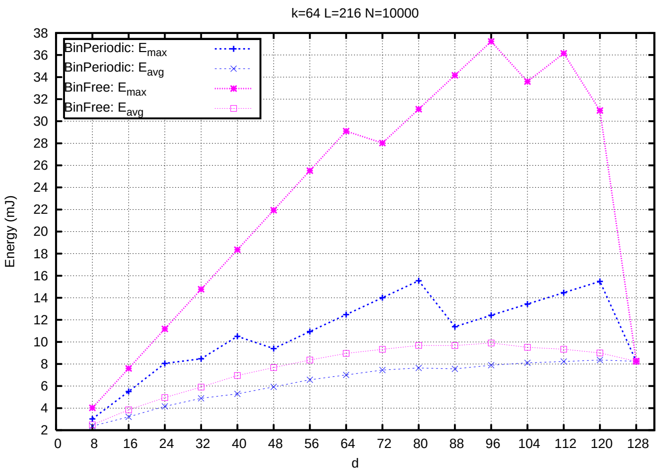

Fig. 15. *Energy consumed when $k = 64$, $L = 216$, and $8 \le d \le 128$.*

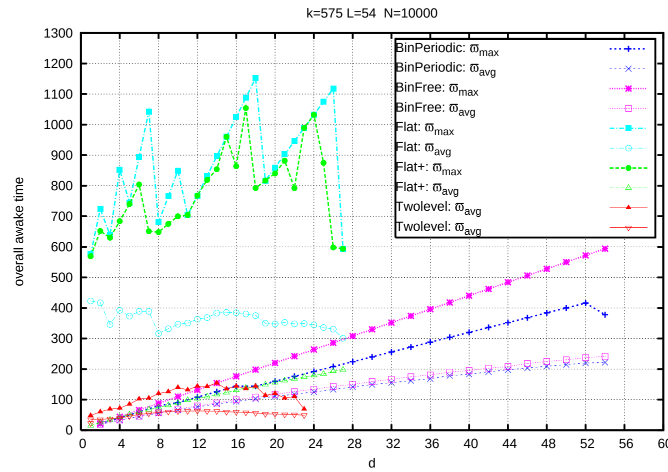

Fig. 17. *Overall awake time slots when $k = 575$, $L = 54$, and $1 \le d \le 54$.*

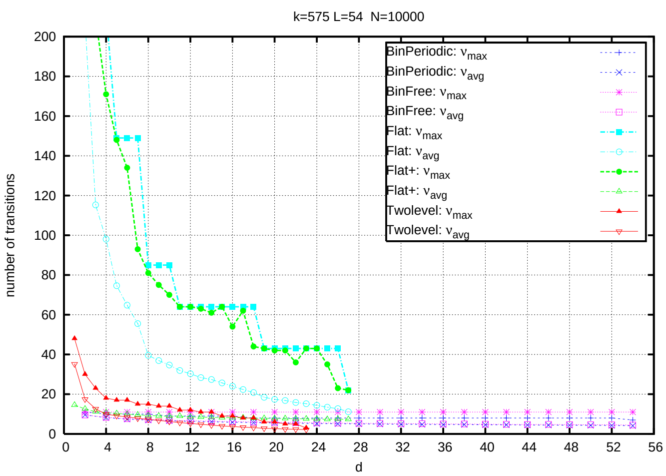

Fig. 16. *Number of transitions when $k = 575$, $L = 54$, and $1 \le d \le 54$.*

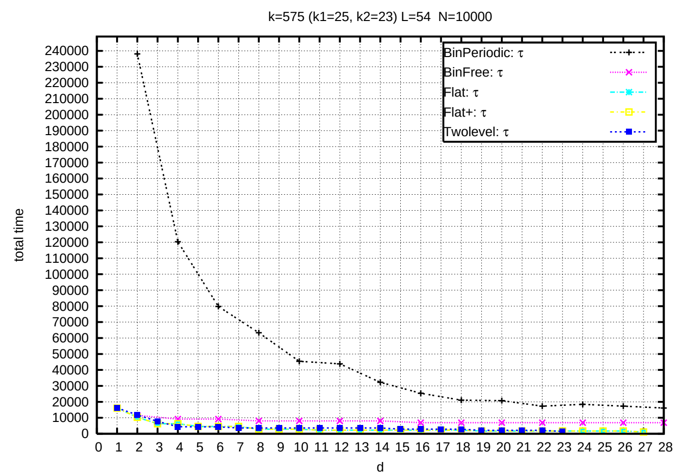

Fig. 18. *Total time slots when $k = 575$, $L = 54$, and $1 \le d \le 54$.*

than the latter. In the worst case, the energy depleted by the Binary-Training protocol is 38 mJ. Since the energy supplied by a sensor is about 4.56 J, the whole training task consumes at most $8/1000$ of the entire energy budget.

In conclusion, a heterogeneous wireless sensor network should use smaller values of d for the free sensors and larger values of d for the periodic sensors. In this way, the BinFree protocol optimizes the overall awake time and the energy consumed, without substantially penalizing the number of transitions, whereas the BinPeriodic protocol optimizes the number of sleep/awake transitions slightly increasing the overall awake time and the energy consumption.

Consider now some experiments where the new Binary-Training protocol is compared with the Flat, Flat+, and TwoLevel protocols, proposed in [1], [2], for homogeneous networks of periodic sensors. In the simulations reported in Figures 16-20, the number k of coronas is fixed to 575, the length L of the sensor sleep-awake cycle is 54 and the sensor awake period d varies between 1 and 54 with a step of 4. The numbers of macro-coronas and micro-coronas for TwoLevel are, respectively, $k_1 = 25$ and $k_2 = 23$, which indeed give $k = k_1 * k_2 = 575$. Note that d is bounded by the length L of the sensor cycle, while for $d = 1$, only the previously known algorithms are defined. In fact, according to Lemma IV.1, Binary-Training requires at least 2 consecutive slots to learn something.

The experiments show how both BinFree and BinPeriodic outperform Flat and Flat+ with respect to $\nu_{\max}$ and $\nu_{\rm avg}$ (Figure 16), and to $\omega_{\max}$ and $\omega_{\rm avg}$ (Figure 17). In particular, for $\nu_{\rm avg}$, although the corona identity range is guaranteed to decrease at each awakening applying both Flat+ and Binary-Training, its range decreases faster using Binary-Training, which guarantees to halve the corona range at each awakening of the sensor. With regard to TwoLevel, its number of transitions is smaller than that of Binary-Training only when d is approximately the same as the number of macro- and micro-coronas. Indeed, when $d = 23$, TwoLevel can train the sensors in just 3 transitions, whereas Binary-Training still uses a logarithmic number of transitions. Clearly, a similar observation holds for the overall sensor awake time.

Concerning τ, Figure 18 shows that the new protocol for periodic sensors is worse than the previous ones when d is very small, confirming that periodic sensors benefit of a moderately long awake period. One can note that, according to Theorems IV.6 and IV.10, BinFree and BinPeriodic have a total time bounded by their number of transitions multiplied by twice the number of coronas and by the Flat total time, respectively.

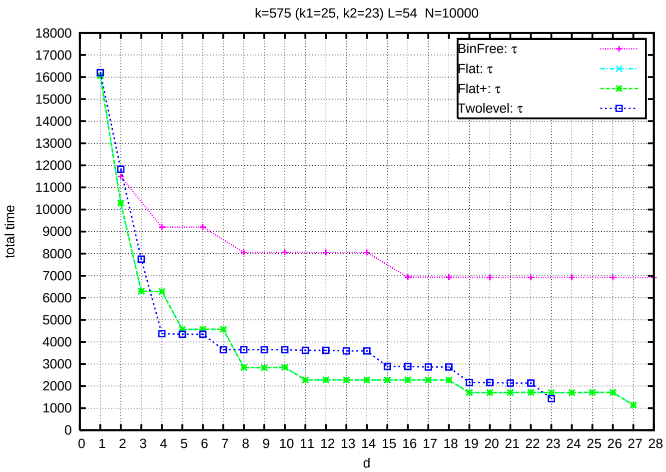

Fig. 19. *Total time slots when $k = 575$, $L = 54$, and $1 \leq d \leq 54$, excluding BinPeriodic.*

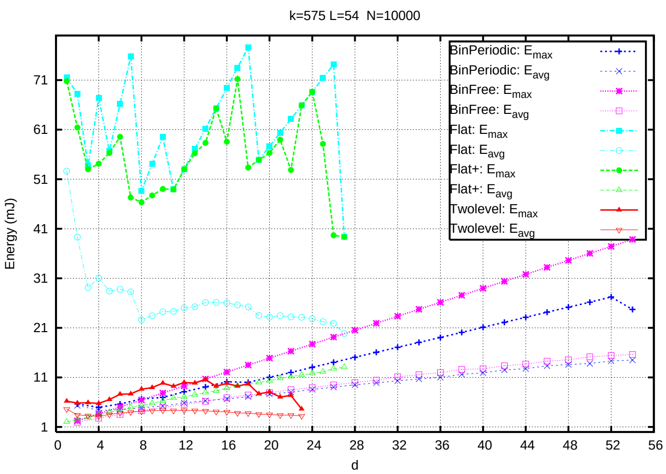

Fig. 20. *Energy consumed when $k = 575$, $L = 54$, and $1 \leq d \leq 54$.*

As shown in Figure 19, BinFree has about a double total time with respect to all the protocols (but BinPeriodic) because $BinFree$ uses both data- and control-broadcasts, and hence in d time slots it hears $\frac{d}{2}$ coronas, while the others hear d coronas. However, the worse time of BinFree is widely counterbalanced by its much lower number of transitions which lead to a moderate energy consumption (see Figure 20). Indeed, BinPeriodic depletes the minimum amount of energy, in both the worst and average cases, with respect to all protocols but TwoLevel. Although TwoLevel has the minimum energy consumption in the average case, it requires a specific actor behavior [1] different from that used by all the other protocols.

The comparison between Flat and Binary-Training for periodic sensors reveals the bicriteria optimization behind a training task: one can either minimize the energy consumption or speed up the training task. Moreover, it is worth noting that in both Flat and Flat+, when the actor transmission is not received, the sensors update the corona identity range deriving from their local time the beacon transmitted by the actor. This makes the Flat and Flat+ protocols very sensitive to slot drifting.

Finally, the above experiments show that Binary-Training for free sensors offers, especially for small values of d, the best compromise for both optimization criteria. Hence, the heterogeneous network takes advantage of the free sensors to become quickly operative, and of the periodic sensors to increase its longevity.

VI. Concluding Remarks

We have proposed training protocols for heterogeneous wireless sensor networks. Heterogeneity comes from the integration of *free* and *periodic* sensors that can independently operate in order to locate themselves with respect to a common powerful device called actor. The actor beacons the network with useful information for localization purposes. The free sensors irregularly alternate between sleep and awake periods, whose frequency and length depend on the protocol computation. Whereas, the periodic sensors alternate between sleep periods and awake periods of predefined lengths, established at the manufacturing time. The analytical and experimental studies have shown that the new protocol outperforms the ones presented in [1], [2], [12] in terms of number of sleep/awake transitions, overall awake time and energy consumption. Moreover, the new protocol is resilient to slot drifting and, to the best of our knowledge, it is the first actor-driven protocol able to train at the same time different types of sensors. The experimental results have also suggested practical choices for the length d of the awake periods: smaller values of d for the free sensors and larger values of d for the periodic ones. As a future work, we also intend to compare the Binary-Training protocol with variants of the synchronous training algorithms, proposed in [3], properly modified so as to tolerate slot drifting.

Acknowledgment

The authors thank D. Borgia for writing the C++ code used in the simulations. Prof. Olariu is funded by the US National Science Foundation under grant CNS-0721563.

References

[1] F. Barsi, F. Betti Sorbelli, R. Ciotti, M.C. Pinotti, A.A. Bertossi, and S.Olariu, Asynchronous training in SANET, in *1st ACM Workshop on Sensor Actor Networks*, Montreal, Canada, September 2007.

[2] F. Barsi, A.A. Bertossi, F. Betti Sorbelli, R. Ciotti, S. Olariu, and M.C. Pinotti, Asynchronous training in wireless sensor networks, in *Proc. 3rd AlgoSensors*, Lecture Notes in Computer Science 4837, 46–57, 2008.

[3] A. A. Bertossi, S. Olariu, and M.C. Pinotti, Efficient corona training protocols for sensor networks, *Theoretical Computer Science*, 402 (1):2–15, 2008.

[4] N. Burri, P. von Rickenbach, and R. Wattenhofer, Dozer: Ultra-low power data gathering in sensor networks, in *Proc. IPSN'07*, Cambridge, MA, April 2007.

[5] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseas, An adaptive power conservation scheme for heterogeneous wireless sensor networks with node redeployment, *IEEE Computer*, 37(8):41–49, 2004.

[6] H. Griffin, *Elementary Theory of Numbers*, McGraw Hill, New York, 1954.

[7] A. Navarra and A. Tofani, Distributed localization strategies for sensor networks, in *Proc. 4th IEEE MASS*, Pisa, Italy, October 2007.

[8] S. Olariu, M. Eltoweissy, and M. Younis, ANSWER: autonomous networks sensor systems, *Journal of Parallel and Distributed Computing*, 67, 114–126, 2007.

[9] S. Olariu, A. Waada, L. Wilson, and M. Eltoweissy, Wireless sensor networks leveraging the virtual infrastructure, *IEEE Network*, 18(4):51–56, 2004.

[10] A. Savvides, L. Girod, M. Srivastava, and D. Estrin, Localization in sensor networks, in *Wireless Sensor Networks*, C.S. Raghavendra, K.M. Sivalingam, and T. Znati, Eds., Kluwer Academic, 2004.
[11] 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.
[12] Q. Xu, R. Ishak, S. Olariu, and S. Salleh, On asynchronous training in sensor networks, in *Proc. 3rd Intl. Conf. on Advances in Mobile Multimedia*, K. Lumpur, Malaysia, September 2005.