HAL open science

Efficient Binary scheme for Training Heterogeneous Sensor Actor Networks

F. Barsi, A. Navarra, Cristina Pinotti, Christian Lavault, Vlady Ravelomanana, Stephan Olariu, A.A. Bertossi

To cite this version:

F. Barsi, A. Navarra, Cristina Pinotti, Christian Lavault, Vlady Ravelomanana, et al.. Efficient Binary scheme for Training Heterogeneous Sensor Actor Networks. 2nd ACM Workshop on Sensor Actor Networks (HeterSANET08), 2008, France. p. 17-24. hal-00461925

HAL Id: hal-00461925

https://hal.science/hal-00461925v1

Submitted on 8 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Binary Schemes for Training Heterogeneous Sensor and Actor Networks

F. Barsi, A. Navarra, C.M. Pinotti
Dept. of Math & Comp. Science
University of Perugia, Italy
{barsi,pinotti}@unipg.it,
navarra@dipmat.unipg.it

C. Lavault, V. Ravelomanana
Lab. d'Informatique de Paris-Nord
University of Paris, France
{lavault,vlad}@lipn.univ-paris13.fr

S. Olariu
Dept. of Computer Science
Old Dominion University, Norfolk, USA
olariu@cs.odu.edu

A.A. Bertossi
Dept. of Computer Science
University of Bologna, Italy
bertossi@cs.unibo.it

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 in support of 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 know training protocols in the number of sleep/awake transitions per sensor needed by the training process. Specifically, in the presence of k coronas, no sensor will experience more than $\lceil \log k \rceil$ sleep/awake transitions and awake periods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HeterSANET'08, May 26, 2008, Hong Kong SAR, China.
Copyright 2008 ACM 978-1-60558-113-2/08/05 ...$5.00.

Categories and Subject Descriptors

C.2.3 [**Computer - Communication Networks**]: Network Operations—*Network Management*; C.2.1 [**Computer - Communication Networks**]: Network Architecture and Design—*Distributed Networks, Wireless Communication*

General Terms

Design Management Algorithms

Keywords

Autonomous sensor networks, sensors and actors, training protocols, free sensors, periodic sensors.

1. 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-lives, service-centric and mission-driven network. This view is in sharp departure from the common wisdom 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 sense 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. Actor-centric sensor networks can detect trends and unexpected, coherent, and emergent behaviors and finds immediate applications to homeland security [8, 5].

As a fundamental prerequisite for self-organization, sensors need to acquire some form of location awareness, see [7, 10]. Almost all applications benefit that the sensed data be supplemented with location information, but not all of them

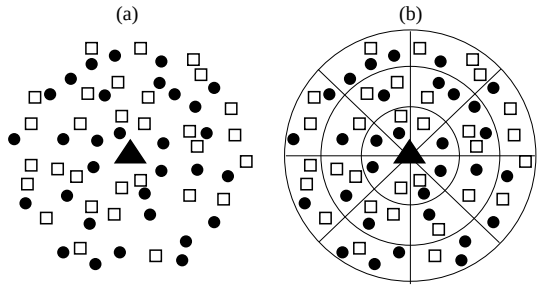

Figure 1: (a) ***A heterogeneous sensor network with a central actor.*** **(b)** ***The trained sensor network. The triangle in the middle represents the actor, circles represent free sensors and squares are periodic sensors.***

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. As illustrated in Figure 1 when the actor transmits, all the sensors awake and belonging to the area covered by the current transmission passively receive the actor's control 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 into equiangular wedges and concentric coronas, centered at the actor, as illustrated in Figure 1(b).

The training protocols reported thus far work on *homogeneous* sensor networks, that is, networks whose sensors are all *identical* in terms of computing, communication capabilities and 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 ad-hoc 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 receiving. 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 in sleep mode a sensor drains much less energy 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 periods of time 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. Both free and periodic sensors wake up, after their deployment, at random. 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 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 process when it wakes up for the first time, without any initial explicit synchronization. It is assumed that, during the training process, both sensors and actor measure 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.

Due to page limitations, in this paper we restrict our discussion to *corona training* that is, the process of acquiring the identity of the corona to which the sensor belongs.

The remainder of this work is organized as follows. Section 2 first discusses the wireless sensor and actor network model and introduces the task of training. In the same section, the actor behavior and sensor protocols along with their worst-case performance analysis are presented. Section 3 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 are then compared with that of previous training algorithms known for the periodic sensors. Finally, Section 4 offers concluding remarks.

2. 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.

2.1 The Network Model

An heterogeneous wireless sensor network is assumed to consist of a single, fixed actor, and 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 use equally long, in phase slots. Each sensor can re-phase its slot with that of the actor every time it wakes up. 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

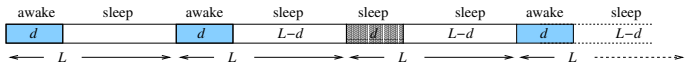

Figure 2: ***The sensor sleep-awake cycle. The darkest d slots represent the time in which a sensor was scheduled to be awake but it decided to maintain the sleep mode due to the expected transmissions from the actor.***

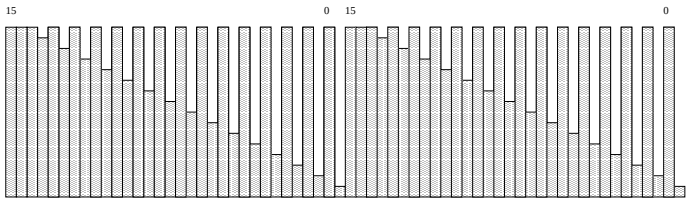

Figure 3: ***Two consecutive actor transmission cycles, with $k = 15$.***

- 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;

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

- The *free* sensors, which alternate between sleep periods and awake periods of arbitrary lengths according to internal computations.
- The *periodic* sensors, which alternate between sleep and awake periods – the sensor sleep-awake cycle is of total length L time slots, out of which the sensor is in awake mode for $d \leq L$ slots, as depicted in Figure 2. Periodic sensors can also sleep for their entire cycle, skipping the awake period.

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 $0 < r_0 < r_1 < \cdots < r_{k-1} = \rho$.

2.2 The Actor Behavior

To train the sensors, the actor repeats a cycle of k *control-broadcasts* at the highest power level, alternated with k *data-broadcasts* at successively decreasing lower power levels (see Figure 3). Since each transmission lasts for a slot, the entire actor cycle lasts $2k$ slots.

As described in Figure 4, at time slots $2i$ and $2i+1$, with $0 \leq i \leq k-1$, the actor broadcasts beacon $k-1-i$, first to all the coronas and then to corona C_{k-1-i}. The actor transmission cycle is repeated for a time τ sufficient to accomplish the training protocol. An evaluation of τ will be given in Theorems 2.7 and 2.10 (for each type of sensor).

It is worthy to note that the redundancy of information between a control-broadcast and the subsequent data-broadcast allows sensors and actor to perform a light synchronization at any time during the training process. Note also that one reason for performing data-broadcast 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 2.5, the inner coronas complete their training earlier than the outer coronas, a subnetwork connected to the actor grows and could start operating before the whole training process terminates.

```
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
```

Figure 4: ***The Binary protocol for the actor.***

2.3 The Sensor Behavior

In order to describe the protocol for 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 in which order (i.e., decreasing, increasing) the data-broadcast transmissions are scheduled.

First the behavior of any sensor, independently of its type, is sketched. Observe that, during the training process, a sensor cannot receive the data-broadcast of any corona identity which is smaller than the corona to which it belongs. Similarly, a sensor can receive the data-broadcast of every corona which is larger than or equal to the corona to which it belongs. To figure out its corona, a sensor uses two $(\lfloor \log k \rfloor + 1)$-bit registers, named *min* and *max*. The *min* register keeps track of the largest corona, discovered so far, smaller than the corona which the sensor belongs to. In contrast, the *max* register keeps track of the smallest corona, discovered so far, which the sensor can belong to. The *min* and *max* registers are initialized to -1 and $k-1$, respectively, because initially each sensor can belong to any corona in $[0, \ldots, k-1]$. From now on, the interval $[min+1, \ldots, max]$ is called the *corona identity range*. From the above discussion, the following *training condition* is verified:

LEMMA 2.1. *A sensor which belongs to corona c, with $c \geq 0$, is trained when* $\max = c$ *and* $\min = c - 1$. ■

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 belongs, each sensor alternates sleep and awake periods. Precisely, after the first awake period, a sensor guesses a corona to which it might belong and goes to sleep until the actor transmits such a corona. At next awakening, after listening the actor transmissions related to the corona it guessed, the sensor reduces its corona identity range. Obviously, the more convenient guess is $\lceil \frac{\min + \max}{2} \rceil$. Indeed, 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 $[C_{\lceil \frac{\min + \max}{2} \rceil + 1}, \ldots, C_{\max}]$. In this way, at each wake/sleep transition of the sensor, the corona identity range reduces at

```
    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|_k:
17                            if  max ≥ first then
18                              max := first; update:=right;
19                              control:= t + i;
20                        first = k:
21                            if min ≤ |c − 1|_k then
22                                min := |c − 1|_k; update:=left;
23                                control:= t + i;
24            t := t + d − 1;
25        if max − min = 1 then
26            mycorona := max;
27            trained := true;
          else
28            guess:= ⌈(max + min)/2⌉;
29            alarm-clock := control + Wait();
30            sleep until alarm-clock rings;
```

Figure 5: ***Training protocol for a generic sensor.***

least by half. The process continues until the range boundaries *min* and *max* differ at most by one. At that point, the sensor is trained and the corona identity, which is the value of the register *max*, is stored in register *mycorona*.

An outline of the sensor behavior is given in Figure 5, where $|a|_b$ stands for the non negative remainder of the integer division between a and b, i.e., $|a|_b = a \bmod b$. In the following, some properties of the training process are discussed.

LEMMA 2.2. *Each sensor requires at least* 2 *consecutive time slots to learn its relative position with respect to the corona identity of the last data-broadcast.*

PROOF SKETCH. If a sensor is awake for just one slot, either it does not receive anything and hence it cannot imply any information, or it may receive an actor transmission but it cannot distinguish among a control- or a data-broadcast.

If a sensor is awake for two consecutive slots, first of all it is sure to receive at least a control-broadcast, hence by receiving or not the previous or the subsequent data-broadcast it can conclude useful information for its positioning by reducing its corona identity range (lines 8-23). □

In the first awake period of the sensor, the corona identity range reduces as follows:

LEMMA 2.3. *Assume a sensor belonging to corona c that wakes up when the actor transmits beacon x, with $0 \le c, x \le k-1$. At the end of the first awake period, the width $\lambda = \max - \min$ of corona identity range of an untrained sensor is:*

$$\lambda = \begin{cases} min\{k-x-1, k-\lfloor \frac{d}{2} \rfloor\} & \text{if } c > x \\ x - \lfloor \frac{d}{2} \rfloor + 1 & \text{if } c \le x \end{cases}$$

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

Thus, by invoking the *Wait* procedure, at every awakening, each sensor intends to receive the corona identity in the middle of its corona identity range (line 25) and depending on its position with respect to such a corona, the sensor will modify either its *min* or *max* boundary. This process is repeated until the sensor becomes trained.

LEMMA 2.4. *In each awake period but the first, every sensor updates only one boundary of its corona identity range unless it becomes trained.*

PROOF. By contradiction consider a sensor that updates both the boundaries and remains untrained. Let *min* and *max* be the values of the boundaries at the beginning of the awake period. During the same 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, at the end of the first awake period, at least $min + 1 - max + k \ge \lfloor \frac{d}{2} \rfloor$ coronas are excluded by the corona identity range. Note that 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 $max - min = 0 - (-1) = 1$. □

Let ν_{max} denote the number of sleep/wake transitions required to train a sensor in the worst case. Applying the binary search scheme on the interval $[min, \ldots, max]$, it holds:

LEMMA 2.5. *A sensor that belongs to corona c and wakes up for the first time while the actor transmits beacon x, with $0 \le c, x \le k-1$, requires to be trained*

$$\nu_{\max} \le \begin{cases} 1 + \lceil \log(\min\{k-x-1, k-\lfloor \frac{d}{2} \rfloor\}) \rceil & \text{if } c > x \\ 1 + \lceil \log(x - \lfloor \frac{d}{2} \rfloor + 1) \rceil & \text{if } c \le x \end{cases}$$

■

In order to analytically evaluate the performance of the *Binary-Training* protocol, in addition to ν_{max}, let ω_{max} be the worst overall sensor awake time, and τ be the total time for training. Recalling that a sensor is awake for d time slots per awake period, one has $\omega_{max} = \nu_{max} d$. Note that τ measures the time required to terminate the whole training process for the actor, whereas each sensor counts in t its *local* training time, that is, how many slots elapse from the sensor

```
    Function Wait: integer;
1     if update = right then
2         Wait := 2⌊(max−min)/2⌋;
      else
3         Wait := 2 (k − ⌊(max − min)/2⌋);
```

Figure 6: ***The* Wait *procedure invoked for the free sensors.***

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 time of all the sensors. The analysis of the total time required by the Binary-Training depends on the *Wait* procedure which determines how long a sensor has to sleep before receiving the *guess* corona.

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

2.3.1 The Free Sensor Behavior

This subsection deals with sensors that can freely choose the 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 6. 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 *Wait* procedure. If *update=right*, *max* is the beacon transmitted at time slot *control* (see Figure 5). Since the *guess* corona is smaller than *max*, *guess* will be broadcasted in the current actor cycle at time slot $control+2\lfloor\frac{\max-\min}{2}\rfloor$. Whereas, if *update=left*, *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 $control+2(k-\lceil\frac{\max-\min}{2}\rceil)$.

Note that the sensor intends to wake up when the actor is transmitting the control-broadcast relative to the *guess* corona. However, the algorithm works properly even if the beacon transmitted is different. Indeed, the sensor updates the *min* and *max* registers listening to the effective actor transmission and does not infer any information from its knowledge of the actor behavior. This makes the protocol reliable with respect to possible drifts on the estimated time.

In order to bound from above the total time τ for the training process, observe that every time the *min* register is modified, the sensor has to wait at most an entire actor cycle, hence:

LEMMA 2.6. *The training process for a free sensor that belongs to corona c cannot last longer than $2k(1+\lceil\log_2 c\rceil)$. Therefore, $\tau < 2k(1+\lceil\log_2 k\rceil)$.* ■

A consequence of the above lemma is that the inner coronas finish the training earlier than the outer coronas. In this way, the wireless sensor network is raised up from the center to the periphery. In conclusion:

THEOREM 2.7. *The free sensors require to be trained $\nu_{\max} \leq (1+\lceil\log_2 k\rceil)$, $\omega_{\max} = d\nu_{\max}$, and $\tau < 2k(1+\lceil\log_2 k\rceil)$.* ■

```
    Function Wait: integer;
1     if update = right then
2         first-corona := |max + d/2|_k;
      else
3         first-corona := min;
4     γ := 1;
5     while guess ∉
      [|first-corona − γ L/2 − d/2 + 1|_k, |first-corona − γ L/2|_k] do
6         γ := γ + 1;
7     Wait := γL − d + 1;
```

Figure 7: ***The* Wait *procedure invoked for the periodic sensors.***

2.3.2 The Periodic Sensor Behavior

In this subsection, the *Wait* procedure for the periodic sensors is devised. For the sake of simplicity, in this section, the parameters d, L are assumed as even, and let each sensor 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 alternates d slots during which it is awake and $L-d$ slots in which it sleeps.

The *Wait* procedure is outlined in Figure 7. Consider a sensor that finishes its current awake period at slot t and invokes the *Wait* procedure. At first, the sensor recomputes in variable *first-corona* the beacon which was transmitted by the actor at the beginning of its current awake period. By Lemma 2.4, if *update=right*, *first-corona* is $|\max+\frac{d}{2}|_k$, whereas if *update=left*, *first-corona* 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 that it has to skip. 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 can receive in every awake period. To analyze the performance of the Binary-Training protocol for periodic sensors, some properties on the coronas received by the sensor are discussed. Denoting with (a,b) the greatest common divisor between a and b, let $L'=\frac{L}{2}$, $\hat{L}'=\frac{L'}{(L',k)}$, and $\hat{k}=\frac{k}{(L',k)}$. Thus:

LEMMA 2.8. *Fixed L, d, and k, and assuming that the sensor wakes up for the first time at slot $2s$, $0 \leq s \leq k-1$, then the corona identity transmitted when the sensor starts the γ-th awake period is $|K_s-\gamma L'|_k = \left|K_s-\gamma(L',k)|\hat{L}'|_{\hat{k}}\right|_k$, where $K_s = C_{|k-1-s|_k}$. Overall only $\hat{k}$ different coronas can be transmitted by the actor when the sensor starts every awake periods, independent of how many sleep-awake cycles the sensor performs.*

PROOF. Consider a sensor that wakes up for the first time at the global time slot $2s$, while the actor is transmitting the beacon K_s. The γ-th sleep-awake cycle of such a sensor starts at time $2s+\gamma L$ while the actor is transmitting the beacon $|k-1-s-\gamma L'|_k = |K_s-\gamma|L'|_k|_k$, with $\gamma \geq 0$ and $L'=\frac{L}{2}$. Observe that L' and k can be rewritten as $L'=g\hat{L}'$ and $k=g\hat{k}$, where $g=(L',k)$. Since $|\gamma L'|_k = g|\gamma\hat{L}'|_{\hat{k}}$ (see [6]), $|\gamma L'|_k$ generates only the $\hat{k}$ multiple of g in $[0,\ldots,k]$, then $K_s-\gamma L' \equiv K_s - \left|g(|\gamma|_{\hat{k}})|\hat{L}'|_{\hat{k}}\right|_k$.

In other words, in any two awake periods, say the i-th and the j-th ones, such that $i > j$ and $i - j < \hat{k}$, the first coronas transmitted are distinct and differ by a multiple of g. Whereas, in any two awake periods i and j such that $i \equiv |j|_{\hat{k}}$ the same coronas are transmitted. □

From the above property, the following result holds:

LEMMA 2.9. *When $\frac{d}{2} = (L', k)$ or $\frac{d}{2} = |L'|_k$, if a sensor stays awake for $\lceil \frac{k}{d/2} \rceil$ consecutive sleep-awake cycles it can receive exactly once all the k coronas identities. Moreover, when $\frac{d}{2} < (L', k)$, not all the sensors can be trained.*

PROOF. The result follows trivially when $\frac{d}{2} = (L', K)$ because, from Lemma 2.8, the $\hat{k}$ coronas transmitted at the beginning of the awake periods divide the k coronas to be broadcast in intervals of (L', k) consecutive coronas and each of such intervals is received by a sensor in a different awake period. Thus, in $\frac{k}{d/2}$ consecutive sleep-awake cycles a sensor receives all the k coronas. When $\frac{d}{2} = |L'|_k$, the sensor receives in consecutive awake periods consecutive coronas and hence all the k coronas are received in $\lceil \frac{k}{d/2} \rceil$ awake periods. □

Since the cycles of the actor and of the sensor last $2k$ and L slots, respectively, then actor and sensors are simultaneously at the beginning of their cycle every $\frac{2kL}{(L,2k)} = \frac{kL}{(L',k)}$ slots. In other words, the cycle of the actor-sensor system, i.e. the minimum time after which both actor and sensors are again in the initial condition, has length $\sigma = \frac{kL}{(L',k)}$ slots. Thus:

THEOREM 2.10. *Fixed L and d even, and k, one has:*
1. *If $d = 2(L', k)$, then $\nu_{\max} \leq 1 + \lceil \log \frac{k}{(L',k)} \rceil$;*
2. *If $d = 2|L'|_k$, then $\nu_{\max} \leq 1 + \lceil \log \lceil \frac{k}{|L'|_k} \rceil \rceil$;*
3. *If $d = 2k$, then $\nu_{\max} = 1$.*

Moreover, $\omega_{\max} = d\nu_{\max}$ and $\tau < \nu_{\max}\sigma$, where $\sigma = \frac{kL}{(L',k)}$.

PROOF. When $d = (L', k)$ and $d = |L'|_k$, by Lemma 2.9, a sensor receives all the corona identities $[0, \ldots, k-1]$ in $\lceil \frac{k}{(L',k)} \rceil$ consecutive sleep-awake cycles. Applying the binary search scheme, since at each awakening it is ensured by the *Wait* procedure and by Lemma 2.4 that the corona identity range at least is halved, one has $\nu_{max} \leq 1 + \lceil \log \lceil \frac{k}{d} \rceil \rceil$. Trivially, when $d = k$, a single sleep-awake cycle is sufficient to train any sensor. By the overall sensor awake time definition, it follows that $\omega_{\max} = d\nu_{\max}$. Finally, since in the worst case a sensor has to wait at most a cycle of the system actor-sensor to receive corona *guess*, the total time of the training algorithm is bounded from above by $\nu_{\max}\sigma = \nu_{\max}\frac{kL}{(L',k)}$. □

3. EXPERIMENTAL TESTS

In this section, the worst and average performance of the corona training protocols are experimentally tested. 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. From now on, the binary training protocols for free and periodic sensors are denoted by *BinFree* and *BinPeriodic*, respectively. Both the worst and the average number of transitions, denoted by $\nu_{\max}$ and ν_{avg}, as well as both the worst $\omega_{\max}$ and the average ω_{avg} overall sensor awake time are evaluated. Moreover, the total time τ, which is the time required to terminate the whole training process, is measured.

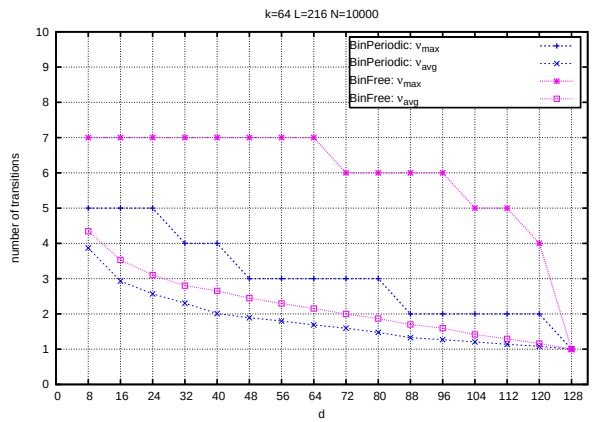

Figure 8: ***Number of transitions when*** $k = 64$, $L = 216$, ***and*** $8 \leq d \leq 128$.

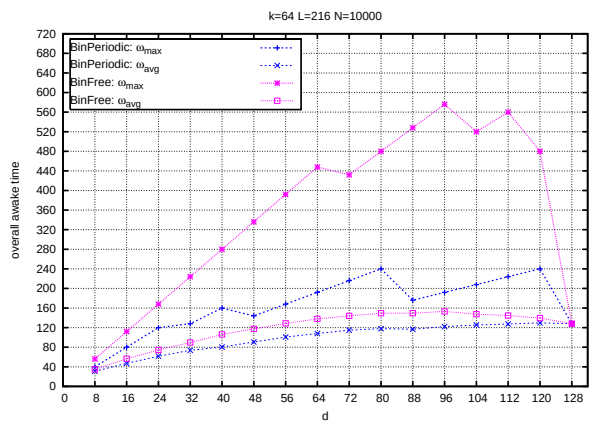

Figure 9: ***Overall sensor awake time when*** $k = 64$, $L = 216$, ***and*** $8 \leq d \leq 128$.

Consider first the experiments for BinFree and BinPeriodic protocols. In the simulations reported in Figure 8, 9, 10 and 11, 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 sensor 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 8 shows the number $\nu_{\max}$ and ν_{avg} of transitions for the different values of d. According to Theorems 2.5 and 2.10, when $d = 2(L', k) = 8$, BinFree and BinPeriodic take $\nu_{\max} = 1 + \lceil \log(k - \lfloor \frac{d}{2} \rfloor) \rceil = 7$ 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. Clearly, increasing d, the gain of BinPeriodic over BinFree increases. As regard to the average performance, although one notes that ν_{avg} considerably improves over $\nu_{\max}$ for both protocols, the improvement is higher for BinFree. Figure 9 presents the awake times $\omega_{\max} = \nu_{\max} d$ and $\omega_{\text{avg}} = \nu_{\text{avg}} d$, which measure, respectively, the worst and average overall energy spent by each sensor to be trained. Clearly, BinFree exhibits awake times larger than those of BinPeriodic since it requires a larger number of transitions. Although the number of transitions decreases as d increases,

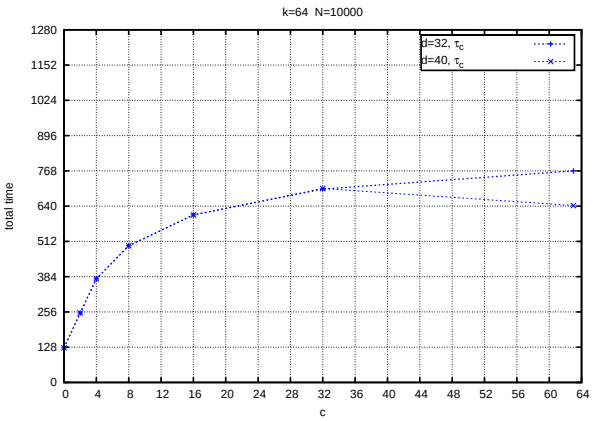

Figure 10: ***Total time required by BinFree protocol to train all the sensors in corona*** $c = 2^i$***, with*** $0 \leq i \leq 6$***, when*** $k = 64$***.***

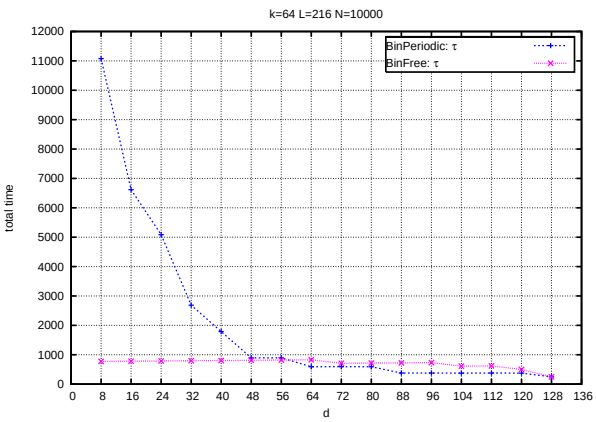

Figure 11: ***Total time when*** $k = 64$***,*** $L = 216$***, and*** $8 \leq d \leq 128$***.***

Figure 9 illustrates that the average overall awake time is constantly slightly increasing for both protocols. 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 $\omega_{\max} = 2\nu_{\max} = 14$. Figure 10 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 2.6, that is $\tau \leq 2k(1 + \lceil \log_2 c \rceil)$. Figure 11 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 $\sigma = \frac{kL}{(L',k)}$ slots, that is a cycle of the actor-sensor system. However, 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 σ to receive the

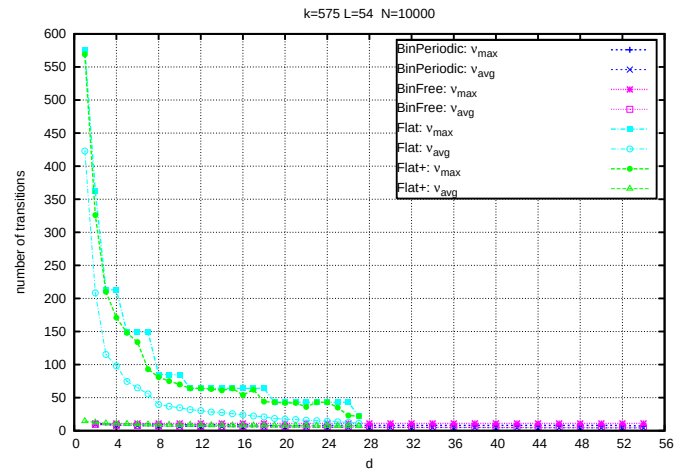

Figure 12: ***Number of transitions when*** $k = 575$***,*** $L = 54$***, and*** $1 \leq d \leq 54$***.***

guessed corona. Note that the total time also decreases because, when d increases, the number of transitions required to train a sensor decreases.

In conclusion, a heterogeneous wireless sensor network should use small values of d for the free sensor and large values of d for the periodic sensors. In this way, the BinFree protocol optimizes the overall awake time without substantially penalizing the number of transitions, whereas the BinPeriodic protocol optimizes the number of wake/sleep transitions increasing tolerably the overall awake time. In the following, the new protocol is compared with the Flat and Flat+ protocols, proposed in [2], for homogeneous networks of periodic sensors. In such protocols, the cycle of actor consists of k transmissions at successively decreasing power level. Each transmission broadcasts the identity of the outmost corona reached. Each sensor repeats the sleep-awake cycle until it is trained, visiting the corona identity range in a peculiar order which depends on the cycle length L and the time s when the sensor wakes up the first time. While Flat does not allow the sensor to skip any awake period, Flat+ does if the corona identities transmitted during a same period are all out of the corona identity range of the sensor. Hence, although the corona identity range is guaranteed to decrease at each awakening applying Flat+, it decreases using either Flat or Flat+ slowly than using the new protocol, which guarantees to halve the corona identity range at each awakening of the sensor. In the simulations reported in Figure 12, 13, and 14, 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. (For $d = 1$, only Flat and Flat+ are defined). Note that d is bounded by the length L of the sensor cycle. The experiments show how the new protocol outperforms both Flat and Flat+ with respect to $\nu_{\max}$ and ν_{avg} (Figure 12), and to $\omega_{\max}$ and ω_{avg} (Figure 13). Concerning τ, in contrast, the new protocol for periodic sensors is worse than the previous ones when d is very small, confirming that periodic sensors benefit of moderately long awake period. The comparison between Flat and Binary-Training for periodic sensors reveals the bicriteria optimization behind a training process: one can either minimize the energy consumption or speed up the training process. Moreover, it is worth noting

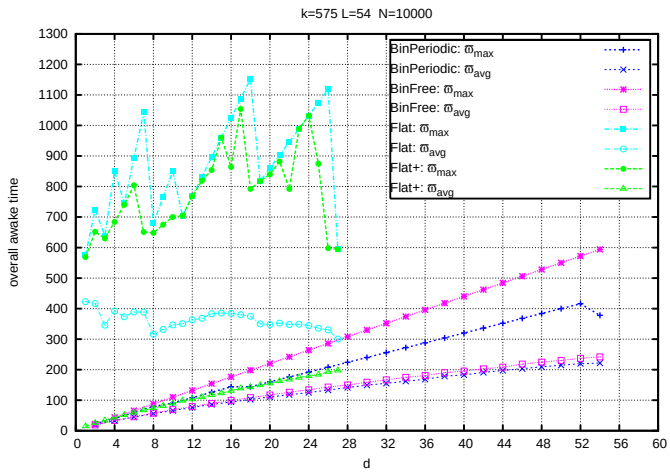

Figure 13: ***Overall awake time when*** $k = 575$***,*** $L = 54$***, and*** $1 \leq d \leq 54$***.***

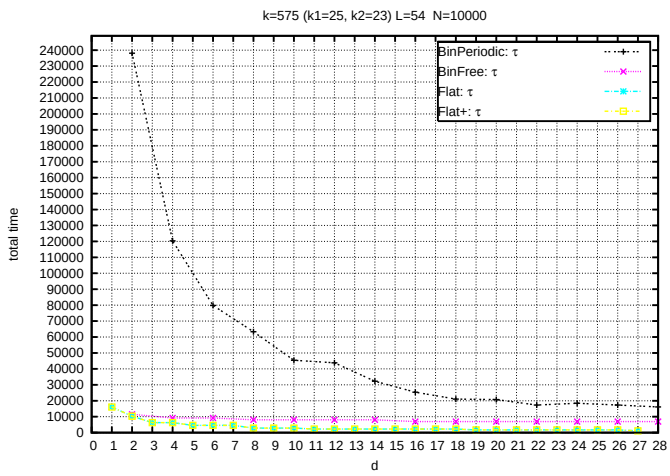

Figure 14: ***Total time when*** $k = 575$***,*** $L = 54$***, and*** $1 \leq d \leq 54$***.***

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 the 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.

4. CONCLUDING REMARKS

We have proposed training protocols for heterogeneous wireless sensor networks. Heterogeneity comes from the integration of *free* and *periodic* sensors that independently can 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 [2] in terms of number of sleep/wake transitions, overall awake time, that is in terms of energy conservation. 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: small 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 the training algorithm proposed in [1].

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.

5. REFERENCES

[1] F. Barsi, F. Betti Sorbelli, R. Ciotti, M.C. Pinotti, A.A. Bertossi, & S.Olariu, "Asynchronous Training in SANET", in *First ACM Workshop on Sensor Actor Networks*, Montreal, Canada, September 10, 2007.

[2] F. Barsi, A.A. Bertossi, F. Betti Sorbelli, R. Ciotti, S. Olariu, & M.C. Pinotti, "Asynchronous Training in Wireless Sensor Networks", in *Proc. 3nd AlgoSensors*, Lecture Notes in Computer Science 4837, 46–57, 2008.

[3] A. A. Bertossi, S. Olariu, and M.C. Pinotti, Efficient training of sensor networks, in *Proc. 2nd AlgoSensors*, Lecture Notes in Computer Science 4240, 1–12, 2006.

[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, A. Tofanir, Distributed Localization Strategies for Sensor Networks, in *Proc. 4th IEEE MASS*, Pisa, October 2007.

[8] S. Olariu, M. Eltoweissy, and M. Younis, ANSWER: autonomous networks sensor systems, *JPDC*, 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 Network, 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, September 2005.