Self-Organizing Sleep-Wake Sensor Systems

K. J. Kwak
Electrical Engineering Dept.
Columbia University
New York, NY 10027
kjkwak@ee.columbia.edu

Y. M. Baryshnikov
Bell Labs
600 Mountain Ave.
Murray Hill, NJ 07974
ymb@research.bell-labs.com

E. G. Coffman
Electrical Engineering Dept.
Columbia University
New York, NY 10027
egc@ee.columbia.edu

Abstract

We propose a self-organizing sleep-wake sensor system that is scalable, easily implemented, and energy conserving. An application of concepts from cellular automata theory accounts for much of its novelty. As a surprising byproduct of its self-organizing behavior, the system has additional, highly desirable properties such as a self-healing capability, fault tolerance, asynchronous operation, seamless accommodation of obstacles in the sensor field, and effectiveness even in the case of intelligent intruders who know sensor design and sensor locations. System performance is a focus of the paper, along with the inverse problem of cellular automata, and self-organizing systems in general: How does one set local rules and initial states so as to achieve pre-specified behavior? Our experimental studies show that broad classes of behavior can be achieved by design, especially by the placement of artificial nucleation centers.

1 Introduction

A low-cost, low-energy sensor deployed in large numbers is a characteristic of many applications of wireless sensor networks. Limited energy is usually accompanied by limited memory, a small sensing range, and a small communication range. To adapt to these constraints, a *sleep-wake protocol* is commonly implemented, whereby sensors alternate between a low-power sleep mode and a relatively high-power wake mode. The system life-time is proportional to the sleep-period duty cycle of this process. A distributed algorithm defined by a local rule coordinates sleep-wake schedules via communications between neighboring sensors within communication range; these communications account for a large part of total energy consumption. Our specific approach focuses on *minimalist self-organizing* techniques, i.e., those making minimal demands on resources, and communication requirements in particular.

In [1, 10], we introduced a class of cellular automata to serve as such a technique. The specific automaton was a generalization of the classical Greenberg-Hastings cyclic cellular automaton on $\mathbb{Z}^2$ to a continuous, but still synchronous version on $\mathbb{R}^2$. In this paper, we make a fundamental extension to an asynchronous automaton, which we shall refer to simply as the asynchronous Greenberg-Hastings automaton, or AGHA, and in so doing we eliminate the extensive communication overhead of distributed synchronization algorithms, both in the initialization stage and in re-synchronization stages created by excessive phase drift or by the redeployment of damaged or expired sensors. By means of extensive experimentation, we illustrate both similarities and certain striking changes in behavior relative to the synchronous GHA. We further illustrate refinements of the techniques for designing AGHAs to meet pre-specified modes of behavior. The planting of *seeds*, or artificial nucleating centers, provides the tool for solving these inverse problems of cyclic cellular automata.

The next section defines a GHA and shows how it can be used in sleep-wake scheduling. This presentation is followed by a brief discussion of relevant background in sleep-wake scheduling. The phase and signaling models are introduced in Section 3. This section also illustrates the dynamics of the asynchronous Greenberg-Hastings Automaton in $\mathbb{R}^2$ under these models. Depending on parameters, striking contrasts can be seen in the wave dynamics; explanations of these effects and the basic differences in synchronous and asynchronous wave dynamics are given in Section 4. Finally, Section 5 presents samples of the results of extensive experiments with a focus on the properties of scalability, fault tolerance, effectiveness against intelligent intruders, and seamless accommodation of obstacles in the sensor field. We conclude by summarizing our findings and by illustrating design of sweep techniques with desirable properties related to intruder detection, forced exit, and entrapment.

2 The Greenberg-Hastings Automaton

Each unit square aligned with the regular lattice, is a *cell*(or *site*) in a cellular automaton; cell x has a discrete value, typically an integer, as its state, and a *neighbor set* N_x defining the cells with which x can "communicate." For example, a commonly studied neighbor set is the von Neumann set consisting of x and the adjacent cells to the north, east, south, and west of x. The state transitions of all cells are synchronized, i.e., they occur at discrete time steps, and they are defined by a universal local rule. Specifically, the transition at cell x at time step t depends on the states of the cells in its neighborhood N_x at time t.

A special class of automata called *cyclic cellular automata* is useful in the design of sleep-wake protocols, and within this class the Greenberg-Hastings Automaton (GHA), defined as follows. Let $\xi_t(x)$ be the state of cell x at time step t. The state transitions of cell $x \in \mathbb{Z}^2$ in the basic GHA follow the local rule:

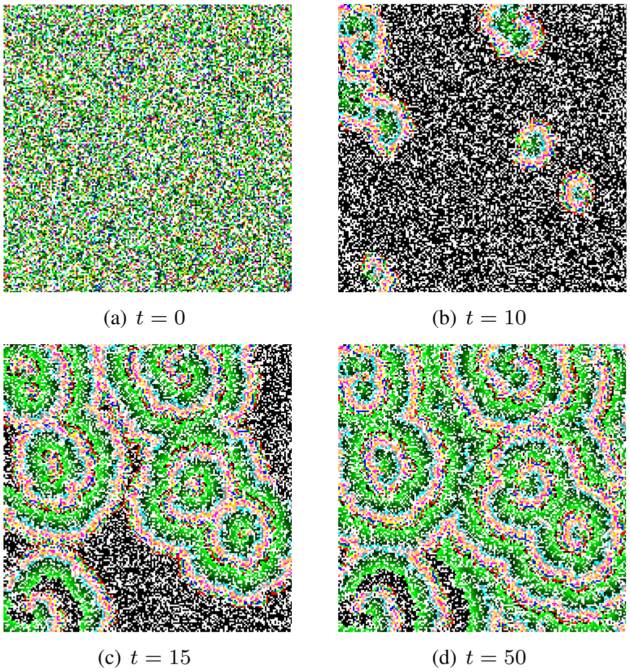
1. If $\xi_t(x) = i > 0$, then $\xi_{t+1}(x) = i + 1 \mod k$.

2. If $\xi_t(x) = 0$ and at least 1 neighbor in N_x is in state 1, then $\xi_{t+1}(x) = 1$; otherwise, there is no change in state: $\xi_{t+1}(x) = 0$.

The state of a cell is said to be incremented *automatically* if it is nonzero, but only *by contact* if it is 0.

Discretized sensor fields are not normally useful models of actual sensor fields; instead, the sensor locations should be modeled as points in $\mathbb{R}^2$, and in the applications (scales) of interest here, point (cell) locations are reasonably taken as i.i.d. uniform random draws from the sensor field, and hence, at the scales of interest here, they will be well approximated by Poisson patterns in two dimensions. Cell x now has a neighborhood N_x defined by all those cells within a disk of given (communication) radius centered at x.

The new GHA model was investigated in [1] and shown to have a periodic behavior very similar to that of the model on $\mathbb{Z}^2$. Figure 1 illustrates typical cellular-automata graphics in which states are mapped one-to-one onto colors, and a finite, but large, square represents $\mathbb{R}^2$. *Unless stated otherwise, default parameters are unit sensor density, and respective communication and sensing ranges of* $r_c = r_s = 1.5$. Snapshots of the evolution of a GHA with 20 states are shown. Bear in mind the convention: *the color black always denotes state 0.* The initial state is a sample from the uniform product measure on $\{0, \dots, k-1\}$. This is commonly called *primordial soup* with reference to models in biology. Emergent behavior begins by a convergence to state 0 (color black), where the cells stay until they find a neighbor in state 1, an event that will eventually occur by the periodic, expanding circular-shaped figures that originate at groups of cells forming *nucleating centers*. For purposes of mathematical analysis, it is useful to "close" the finite field by joining opposite edges. Here, however, we shall just ignore edge effects, as the scale of our experiments and the applications of interest is large enough to make these effects relatively negligible.


(a) $t = 0$

(b) $t = 10$

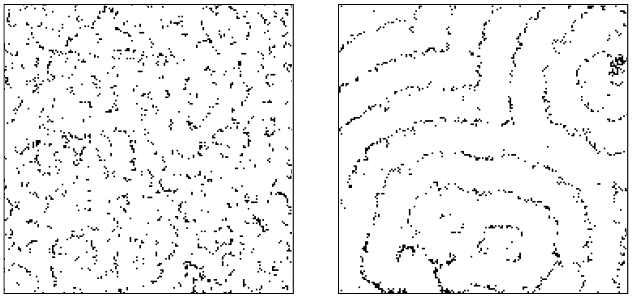
(c) $t = 15$

(d) $t = 50$

Figure 1. Greenberg-Hastings Automaton in $\mathbb{R}^2$ (k=20)

Turning to our sensor application, point-cells are now sensors; state 0 represents the *wake* state in which the sensor is actively sensing events within sensing range; and state 1 is the state in which a sensor signals all sensors within communication range that it is in state 1, so that those neighbors in state 0 can transition into state 1. States 2 through $k-1$ are called *sleep* states since no sensing or communication takes place in these states. Figure 2 shows just the sensors in state 0 at $t = 100$ with $k = 12$ and $k = 20$. They form many small, generally curved line segments when $k = 12$ as in Figure 2(a) but they form the closed periodic waves when $k = 20$ as in Figure 2(b). In the appendix Figure 13 helps visualize wave dynamics by giving closely spaced snapshots of the state. Note the important fact that communication, *the high-energy sensor function, can be turned off completely once stable periodic behavior is reached,* which happens quickly for the designs of practical interest.

With reference to Figure 2(b) consider the occurrence of an event (a fire, appearance of an intruder, ...) at some point of the sensor field. It will be detected (i.e., sensed) immediately if it occurs within sensing range of a sensor



(a) k=12

(b) k=20

Figure 2. Greenberg-Hastings Automaton in $\mathbb{R}^2$ ($k = 12, 20$)

in state 0. Otherwise, it will be sensed as soon as the next wave of wake sensors gets within sensing range. This may never happen if the wake sensors have large enough gaps between them, but designs where detection is almost certain to happen are easily found by taking the sensor density or sensing range large enough. Of course, if the "event" can move, then it can avoid detection, but eventually it must move out of the sensor field. We cover these details in later sections.

Nucleating centers, i.e., natural seeds, play a key role. If k is chosen too large (greater than about 26) then from primordial soup the GHA will *fixate* almost certainly with all sensors permanently in the wake state, i.e., state 0 (the graphic will be all black). In this case there is no natural nucleation in the emergent behavior of the GHA. Classical results show an exponential scaling of the probability of fixation relative to the number, k, of states and the communication area defining neighbor sets. For purposes of the sensor-system application, our interest focuses on values of k that are indeed beyond those for which nucleating centers can be expected to form naturally. In this case, artificial seeds can be planted as a natural, easily implemented extension of the GHA. This technique, introduced in [1], is covered in Section 4 and greatly extended in subsequent sections, especially for intruder entrapment.

Background. The literature dealing with sleep-wake protocols is large, both for synchronous systems [9, 16, 21, 3, 12, 14, 15, 11, 19] and asynchronous [18, 20, 17, 4, 13, 7] systems. These protocols can be classified by general objectives and by the constraints under which they must operate. For example, there exist protocols requiring that link connectivity be preserved [21, 13] and others that require full, or nearly full, coverage of the field at all times [9, 16, 21, 12, 14, 18, 17, 7, 4, 19], and yet others that constrain the delivery of sensing data to centralized clearing points [3]. The techniques used vary from simple randomization methods [9, 13, 7] to LP-formulations of sleep-schedule optimization [3, 11] and the domatic partition approach [12, 14]. As is to be expected, the implementation complexity of optimization techniques is substantially, and in applications of interest here, prohibitively greater than that of a minimalist cellular automaton.

Some protocols [21, 18, 20] implement wake-sensor density control, while others [15] implement "sweep" protocols in much the same spirit as the technique proposed here but much more demanding of resources, i.e., much less of a minimalist protocol for maximizing system lifetime. Wake-sensor wave-propagation provides a flexible and systematic trade-off between energy consumption and time-to-detection or entrapment.

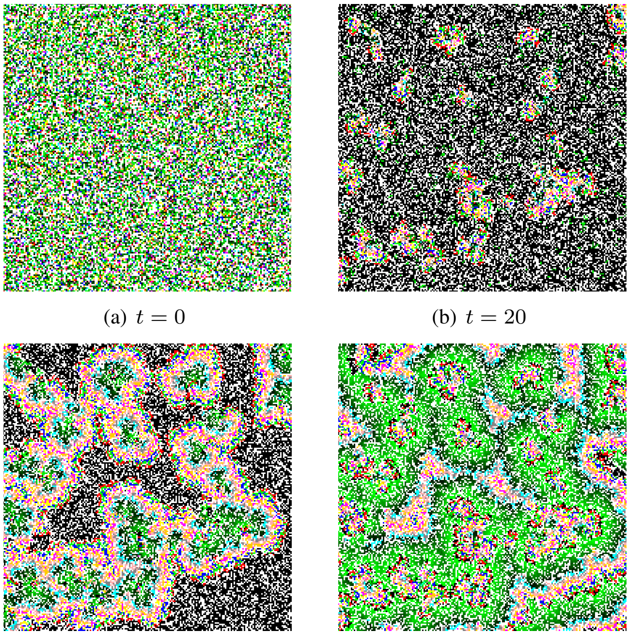
Many Self-organizing schemes were introduced for sleep scheduling. They are similar in that those schemes use distributed algorithm to achieve self-organization. However most schemes are different from our scheme in that they are based on the clustering or they consider full coverage of wake sensors. Note that our scheme is flat protocol and provides partial coverage with guaranteed detection delay. Many references can be found in [8, 2, 5, 6].

3 The Asynchronous Greenberg-Hastings Automaton

3.1 Phase and Signaling Models

The sensors in the Asynchronous GHA (AGHA) share the same local rule as in the synchronous case; and they are controlled by the same parameters, i.e., the number of states, the communication radius, the sensing radius, and the sensor density. We adopt the restricted model of asynchronous systems in which sensor-clock cycles have constant durations, but differ in their relative phases. We adopt a standard, baseline probability model: The phases of the sensor clocks are chosen independently and uniformly at random from a discrete set; these sets will be varied and will be stipulated in due course. To keep simulation state spaces small (and experiments not too time-consuming), the size of these sets is kept as small as possible consistent with their purpose, i.e., the size is not taken so small that the properties being studied become unclear or ambiguous.

Compared to synchronous sensor systems, the nature of the partial overlapping in clock cycles amongst neighbors plays a crucial role in shaping the wake-sensor waves, since successful communication must be made during the overlaps. We propose two different signaling techniques, with different energy requirements, to investigate how clock phase affects the dynamics and performance of a sensor system. Where sensing involves signal detection, we assume that the sensing and the communication devices do not interfere with each other (e.g., their frequency spectra could be well separated), so we may suppose that sensors



(a) $t = 0$

(b) $t = 20$

(c) $t = 30$

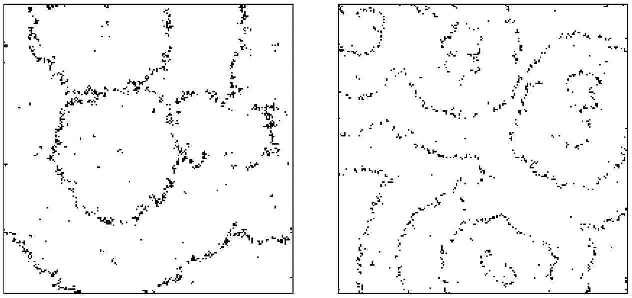
(d) $t = 60$

Figure 3. Asynchronous Greenberg-Hastings Automaton in $\mathbb{R}^2$ (k=20)

in state 0 can turn on their radios (communication devices) at any time in the clock cycle while sensing devices are on.

Our first proposed signaling technique is one in which a sensor in state 1 broadcasts a signal, a short low-energy burst consistent with the communication range, at both the beginning and end of the clock cycle. This *type-1* clock structure increases the number of communication signals by a factor of two, but it also increases the chance of successful communication when there are large phase differences; in this sense it is a best-case assumption relative to "catching" a neighbor in state 1, but a worst-case assumption in terms of energy dissipation (in that two signals are used instead of just one).

Our second scheme for comparison is simpler and uses less energy: a sensor in state 1 broadcasts a signal in the middle of the clock cycle. Adopting this *type-2* scheme decreases the number of communications, keeping signaling at a minimum, but the needed overlap duration becomes larger, i.e., as in the previous section, at least half of a clock cycle must be overlapped to make a successful communication.



(a) Type-1 Clock Structure

(b) Type-2 Structure

Figure 4. AGHA with Different Clock Structure (k=20)

3.2 Experiments

We begin with a coarse set of possible overlaps: Assume that the phases of the sensor clocks are chosen so that fractional clock-period overlaps are independently and uniformly distributed on $\{0, 1/5, 2/5, 3/5, 4/5\}$.

Figure 3 illustrates the sequential evolution of the AGHA with 40,000 points placed independently and uniformly at random within a 200×200 field. The process is started in primordial soup with $k = 20$ and $r_c = 1.5$, and type-2 signaling is used. We note that, at this point, there are no striking differences; similar to the (synchronous) GHA, most cells experience a delay in state 0 waiting for a neighbor in state 1; see Figure 3(b) for an illustration of the asynchronous case. Once a neighbor in state 1 appears, cells start to generate periodic circular wave patterns (or patterns corresponding to the periphery of intersecting circles) just as the synchronous GHA did. see (Figure 3(d)).

Next, consider the type-1 "double" signaling scheme. The snapshot of equilibrium in Figure 4 reveals an interesting property of type-1 signaling: the width of a wake-sensor band is enlarged, and the gaps between successive waves are correspondingly enlarged. However, as noted earlier the clock-cycle k-periodicity is preserved, as it must be. Thus, the broader wake-sensor bands propagate faster than in the (synchronous) GHA, so all sensors still come awake every k-th clock cycle. A more detailed explanation of wake sensor bands and their variation with parameters is given in the next section. Inspection of Figure 13 shows that the type-2 asynchronous sensor system differs little from the synchronous case. We also explain this somewhat unexpected phenomenon in the next section.

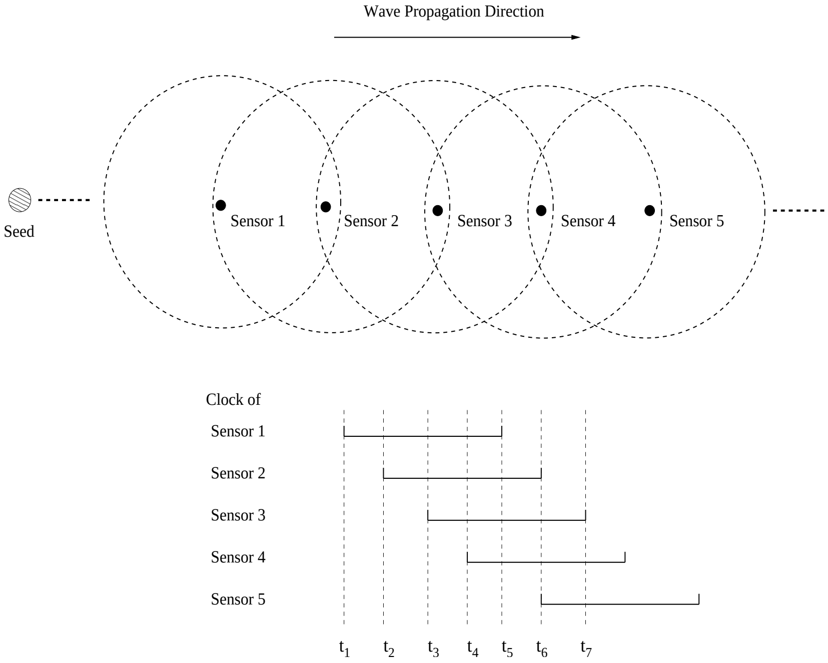


Figure 5. Wake Sensor Band

4 Detailed Wave Dynamics

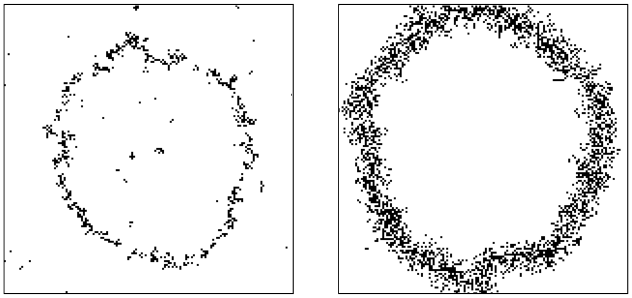
Figure 5 illustrates the conditions under which wide wake-sensor bands are formed, when we assume type-1 signaling and large k (the system fixates without seeding). The figure considers the sensors in a regular, one dimensional array orthogonal to the direction of wave motion, as this framework greatly simplifies the discussion without obscuring the effects to be brought out. Initially, prior to time t_1 in the figure, all sensors fall into state 0 and wait for a state-1 signal to propagate from a seed. At time t_1, sensor 1 responds to a state-1 signal received in the previous clock cycle, increases its state to 1, sends out its first state-1 signal, and remains in state 1 until t_5. At t_2, sensor 2 responds to the state-1 signal from sensor 1, transitions to state-1, sends out its first state-1 signal and remains in the 1 state until t_6. This process continues until sensors 1 through 4 are all in state-1 for the last quarter clock cycle of sensor 1; at this point of the wave dynamic, the width of the wave is created by 4 sensors. The state as a function of time is summarized in Table 1.

For greater widths, we need greater densities, as suggested by Figure 6. Figure 6 illustrates how density affects the widths of wake-sensor bands. The parameters are $k = 30$, a communication radius of 1.5, sensors are distributed uniformly at random over a 150×150 sensor field, there is a centrally located artificial seed, and initial (fractional) phases are selected uniformly at random from $\{0, 1/40, \ldots, 39/40\}$. Results are shown for sensor densities 1 and 4. The wavefronts for the higher density are much thicker, as is immediately obvious. The waves are far apart (too far apart to show two of them on the same figure for density 4) and move correspondingly faster in keeping with the local k-periodicity. Interestingly, this method of producing thick and dense wave fronts provides another approach to ensuring high-security blocking of mobile intruders.

time	1	2	3	4	5
$[t_1, t_2)$	1	0	0	0	0
$[t_2, t_3)$	1	1	0	0	0
$[t_3, t_4)$	1	1	1	0	0
$[t_4, t_5)$	1	1	1	1	0
$[t_5, t_6)$	2	1	1	1	0
$[t_6, t_7)$	2	2	1	1	1
	⋮	⋮	⋮	⋮	⋮

Table 1. Sensor States

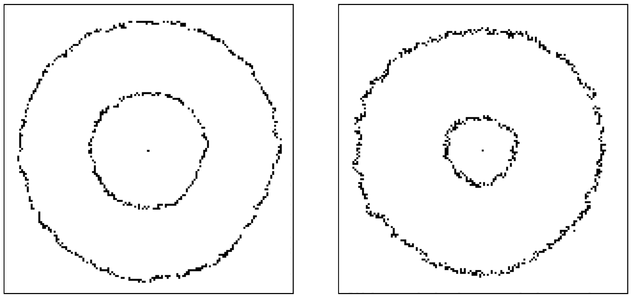
Returning to the similarity between the behavior of synchronous and type-2 asynchronous systems, consider the comparison available in Figure 7, where the sensor field is 150×150, the density is 2, $k = 30$, $r_c = 1.5$ and there is a single center seed. Phases (fractional overlaps) are multiples of 1/5. Recall that the synchronous system evolves from primordial soup in three stages: a *reset* stage in which



(a) Density=1

(b) Density=4

Figure 6. Wake Sensor Band with Different Node Density



(a) Synchronous System

(b) Asynchronous System

Figure 7. Comparison Between Synchronous and Asynchronous System. (1 single-phase seed at the center of field with k=30 and density=2.0.)

almost all sensors end up in state 0 waiting for a state-1 signal to propagate from the seed, a *stabilization* stage during which the wave dynamics are developed, and finally an *equilibrium* stage in which practically all sensors are effectively independently periodic in a global, deterministic process. For any sensor, approximate the number of its neighbors with a phase overlap sufficient for communication by a Poisson random variable with mean given by the product of the density, the communication area, and the fraction of phases giving sufficient overlap for successful communication between two neighbors: $2 \cdot \pi r_c^2 \cdot 0.6 = 8.482$. Thus, the probability that there are no such neighbors is 2×10^{-4}. As a crude estimate, one can expect the fixation rate to be well below one in a thousand. By our earlier arguments, the thickening of the wave fronts established earlier in this section can be expected to be perceptible for type-2 asynchronous systems only for sufficiently large densities, as in the figure where the density is 2. For densities of 1 or less, the increased width becomes imperceptible. By this informal argument then, one should expect that the synchronous and the type-2 asynchronous wave dynamics will look roughly the same with a small thickening of the wave front for high densities.

As in the synchronous case, for energy conservation, one prefers large k. The locations of nucleating centers (seeds) will be unpredictable as will be the wake-state wave action they induce, but good performance is assured. Note that fixation is in state 0 so that there will be no sacrifice in surveillance, but the lifetimes of the sensors will be reduced. On the other hand, in the interests of low energy consumption and hence a low duty cycle $1/k$, we will want to take k large. And k does not have to be very large for practical field sizes ($k > 25$ will do when we have density 1, $r_c = 1.5$ and a 100×100 field). *Planting seeds*, i.e., deploying artificial nucleation centers, is a handy and effective solution for larger k. Any collection of sensors containing a $k - cycle$ serves as a seed. As the term suggests, a k-cycle in the asynchronous sensor system is a sequence of sensors $x_0, \ldots, x_{k-1}$, who are synchronized each other, such that, for all $k = 0, \ldots, k-1$, $\xi_t(x_i) = i$ and $x_{(i+1) \bmod k}$ is in communication range of x_i. Clearly, a k-cycle, which cycles endlessly through the k states, spending one clock period in each state, is trivial to put together, and renders fixation impossible

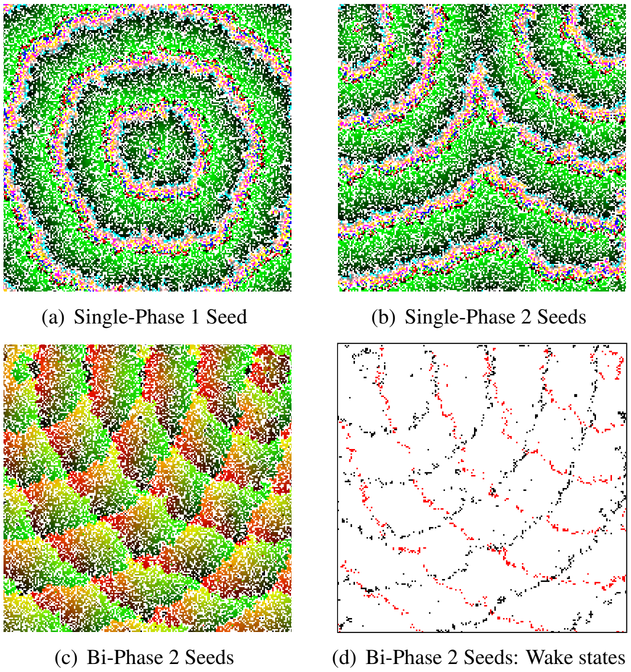
Implementations of this technique can be the same as in synchronous system, and performance will remain unchanged except for the wave speed-width details as described above: thicker, faster waves, but with k- periodic equilibria remaining an invariant. Figure 8 illustrates the new but very similar dynamics. Figure 8(a) and 8(b) shows the dynamics of system at time step 200 with $k = 30$ when 1 and 2 artificial seeds with the same state space are planted and function independently.

To enhance protection against intruders, or to speed up detection, the use of *bi-phase* sensors in a two-seed asynchronous system is a technique worth considering. One obtains results like those in Figure 8(c) and 8(d). Recall that the sensors are now designed to maintain two out-of-phase AGHAs simultaneously, which means a (near) doubling of the wake-sensor duty cycle.

5 Experimental Results

As pointed out earlier, the expected delay until detection, $\mathbf{E}D$, is the same as in the synchronous case (because of the invariant k-periodicity of a stabilized system). In [1] it is shown that

$$\mathbf{E}D \approx \left(1 - \frac{2r_s}{kr_c}\right) \frac{kr_c - 2r_s}{2r_c}$$


(a) Single-Phase 1 Seed

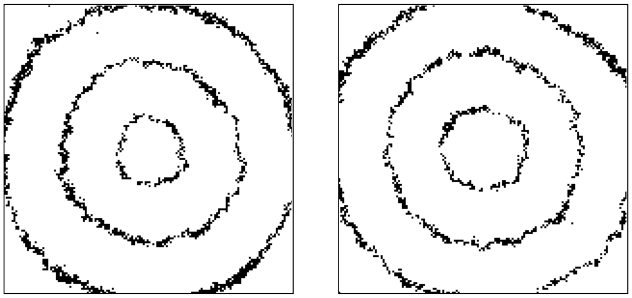
(b) Single-Phase 2 Seeds

(c) Bi-Phase 2 Seeds

(d) Bi-Phase 2 Seeds: Wake states

Figure 8. Planting artificial seeds with k=30

Experiments for the asynchronous system with a center seed gave Table 2, which shows, as in the synchronous case, excellent agreement between the experimental results and the conservative analytical estimate.


(a) $p = 0.1$, density=2

(b) $p = 0.3$, density=2

Figure 9. Link Failures in Asynchronous System. (Single seed with k=30)

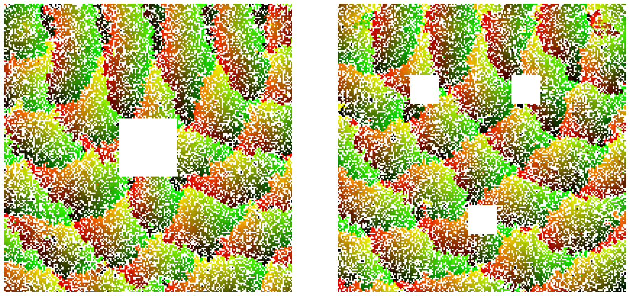
We tested the asynchronous system with a stationary link-failure probability p to illustrate how link failures affect system dynamics. Link failure probabilities $p = .1, .3$ are considered in Figure 9. As p increases, the wake-sensor wavefronts sustain more "hollows" owing to failures to successfully receive broadcasts from neighboring sensors in state 1. The effects on performance of the AGHA are practically the same as on the GHA, although it should be noted that wake sensors in the AGHA can receive more state-1 signals from neighboring sensors, no matter where the latter are relative to the direction of wave motion; they can be ahead or behind – only the relative phases matter. It is clear from the figures that substantial robustness in the presence of link failures requires substantial sensor densities.

Seamlessly accommodating obstacles is another property of the GHA that is preserved in the AGHA. To confirm this fact, we again experimented with one big obstacle (40×40) and three small obstacles (20×20) in a 200×200 sensor field. Figure 10 shows that, as in the synchronous system, our proposed scheme can gracefully work around both the one huge obstacle and the small obstacles; the system continues to pump out periodic waves sweeping the area outside the obstacles; the wake sensors sweep the obstacle along its boundary and close up the waves beyond the obstacle.

As stated earlier, one advantage of the asynchronous system is that sensors can be redeployed without a costly re-synchronization in an area where sensors have expired through damage or old age. The operation of newly de-

k	Clock Structure 1	Clock Structure 2	Estimates
15	6	7	6
20	8	8	8
25	10	11	11
30	13	13	13
35	16	15	16

Table 2. Estimates of Average Detection Time (in clock cycles)


(a) One Obstacle - All sensors

(b) Three Obstacles - All sensors

Figure 10. Working Around Obstacles (k=30)

ployed sensors is identical to that of sensors in the initial phase. Newly deployed sensors increase their states until they reach state 0 and wait for a state 1 signal. Once the new sensors receive this signal, they increase their state to 1 and subsequently become a part of the periodic wake-sensor waves. Figure 11 shows some snapshots of system behavior when sensors are redeployed on a small scale. In the overall field, when it was initialized, 40,000 sensors were deployed in 200×200 field with $k = 40$. One artificial seed is planted at the center of the sensor field and wake sensors (in state 0) are displayed in black. Figure 11(a) shows the sensor system as it stabilizes in periodic waves. As illustrated in Figure 11(b), assume certain sensors die (identified as red points with arrows)in two square areas, one in the upper-left and one in the lower-right region of the field. These *inert* areas can be created by external forces such as local flooding, crushing by vehicles, etc. As sensors die, the sensor system adapts itself to the new configuration and begins a self-healing process. Wake sensor waves sweep along the boundary of the inert areas and keep propagating wake states without major distortions in the dynamics, as in Figure 11(c). To recover the surveillance in the inert areas, new sensors are deployed at random; these eventually reach state 0, and wait for a state-1 signal, as in Figure 11(d). As wake-sensor waves arrive, new sensors start to catch state-1 signals, coalesce with incoming waves, and knit together a new wave dynamic much like the original. This self-organizing process is complete around 50 clock cycles after redeployment, as shown in Figure 11(f).

This above illustration is typical of small areas of sensor damage, but as the areas become rather large, the self-organized knitting together of new structures can introduce rather chaotic wave formations, depending on area size and the relative position of the damaged area and the field's artificial seeds. Thus, for larger areas, it is best simply to reset the entire sensor field, so that it regenerates a regular wave dynamic. This reset control will be part of the centralized re-deployment mechanism

6 Conclusion

We have presented a self-organizing, asynchronous sleep-wake protocol for sensor systems that are low-cost, self-healing and long-lived without sacrificing essential properties like fault tolerance and smooth accommodations of obstacles. The method yields an attractive mechanism for trading off system life-time with detection delay. We have shown by comparisons with results in [1] that asynchronous systems are at least as efficient as synchronous ones and have properties, such as wave thickening, that actually improve on performance. By appropriate placement of seeds in systems that otherwise fixate, many different wave patterns may be explored. Against the circular waves of the center-seed system, intruders can not penetrate the field "on the ground" beyond a short distance from the boundaries. An intruder dropping into the field at any location away from regions near the boundaries is forced to exit staying between consecutive waves if he is to remain undetected.

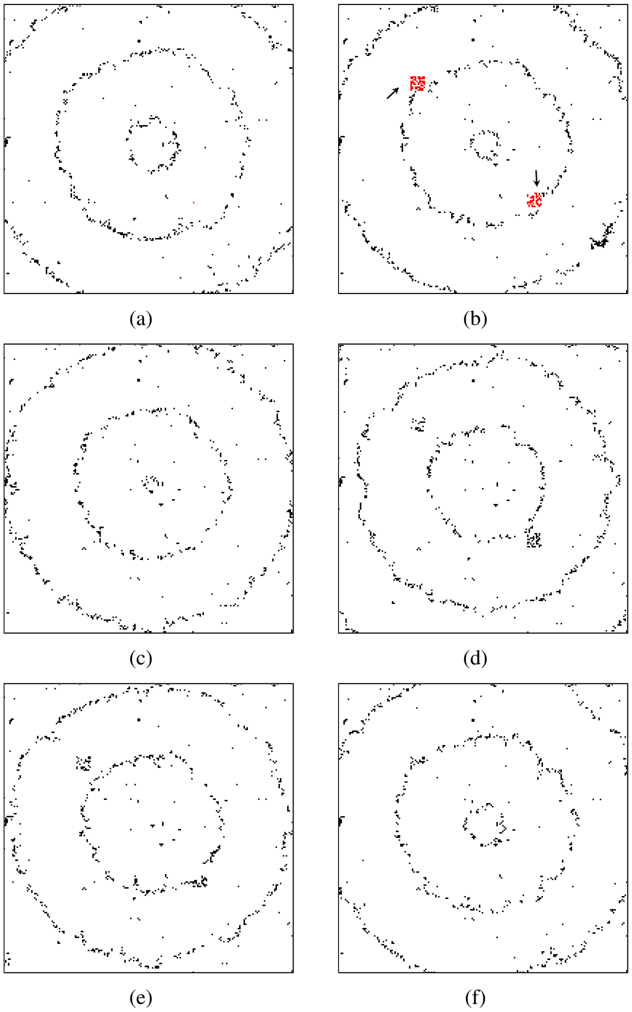
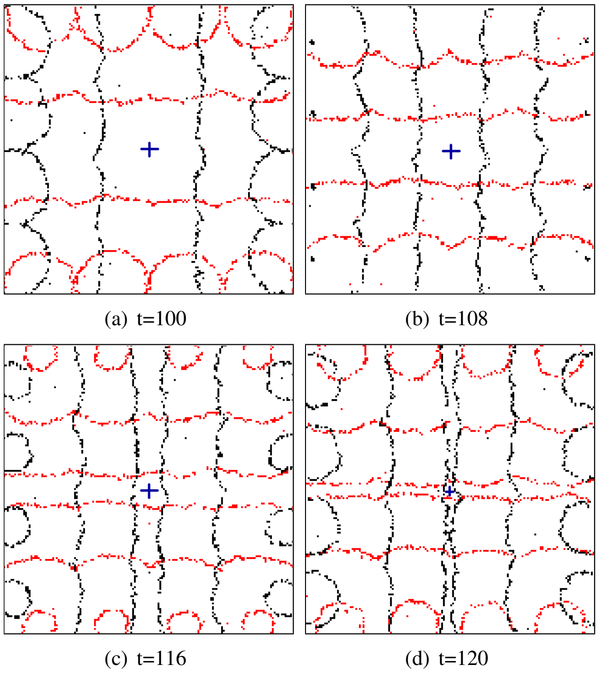


Figure 11. Sensor Redeployment with k=30

A state of a total entrapment protocol is illustrated in Figure 12. In the figure, waves are moving up from the bottom and down from the top; while orthogonal waves are moving similarly in a different phase from right to left and left to right. Points of intrusion like the one shown are trapped within a quadrilateral formed by four approaching waves. The intrusion is shown to take place at the center of the field. If it had been chosen off center and the intruder wished to delay entrapment as long as possible, he would have moved toward the center at an appropriate speed, whereupon the same entrapment event would have occurred as shown in the figure. Some readers might worry about the intrusion coming toward the center of sensor field at the border. To prevent such intrusions and protect a certain area, we place an artificial seed at that area, as in 8(a), such that circular wake sensor waves propagate toward boundaries from that area and intruders will be pushed away.


(a) t=100

(b) t=108

(c) t=116

(d) t=120

Figure 12. Entrapment: Bi-phase, 8 seeds for each phase, are planted over a 160×160 sensor field with *k=25*

References

[1] Y. M. Baryshnikov, E. G. Coffman, and K. J. Kwak. High performance sleep-wake sensor systems based on cyclic cellular automata. In *Proceedings, IPSN '08*, 2008.

[2] P. K. Biswas and S. Phoha. Self-organizing sensor networks for integrated target surveillance. *IEEE Transactions on Computers*, 55(8):1033–1047, 2006.

[3] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of IEEE INFOCOM '02*, pages 1278–1287, 2002.

[4] B. CHen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[5] Z. Chen and A. Khokhar. Self organization and energy efficient tdma mac protocol by wake up for wireless sensor networks. In *Proceedings of IEEE SECON '04*, pages 335–341, 2004.

[6] Falko Dressler. *Self-organization in sensor and actor networks*. Wiley, January 2008.

[7] C. Hua and T. P. Yum. Asynchronous random sleeping for sensor networks. *ACM Transactions on Sensor Networks*, 3(3):15–25, 2007.

[8] A. Patel J. Degesys, I. Rose and R. Nagpal. Desync: self-organizing desynchronization and tdma on wireless sensor networks. In *Proceedings of ACM/IEEE IPSN '07*, pages 11–20, 2007.

[9] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *Proceedings of MobiCom' 04*, pages 144–158, 2004.

[10] Kyung Joon Kwak. *Minimalist Detection and Counting Protocols in Wireless Sensor Networks*. PhD thesis, Columbia University, Electrical Engineering Department, October 2008.

[11] S. Meguerdichian and M. Potkonjak. Low power 0/1 coverage and scheduling techniqus in sensor networks, Jan 2003.

[12] T. Moscobroda and R. Wattenhofer. Maximizing the lifetime of dominating sets. In *Proceedings of the IEEE International Workshop on Algorithms for Wilress, Mobile, Ad Hoc and Sensor Networks*, page 242b, 2005.

[13] V. Paruchuri, S. Basavaraju, R. Kannan, and S. Iyengar. Random asynchronous wakeup protocol for sensor networks. In *Proceedings of IEEE International Conference on Broadband Networks (BROADNETS '04)*, pages 710–717, 2004.

[14] S. V. Pemmaraju and I. A. Pirwani. Energy conservation via domatic partitions. In *Proceedings of MobiHoc '06*, pages 143–154, 2006.

[15] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang. Design and analysis of sensing scheduling algorithms under partial coverage for object detection in sensor networks. *IEEE Transactions on Parallel and Distributed System*, 18(3):334–350, March 2007.

[16] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless snensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 02)*, pages 32–41, 2002.

[17] X. Wang, G. Xing, G. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys '03)*, pages 28–39, 2003.

[18] K. Wu, Y. Gao, and F. Liand Y. Xiao. Lightweight deployment-aware scheduling for wireless sensor networks. *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks*, 10(6):837–852, 2005.

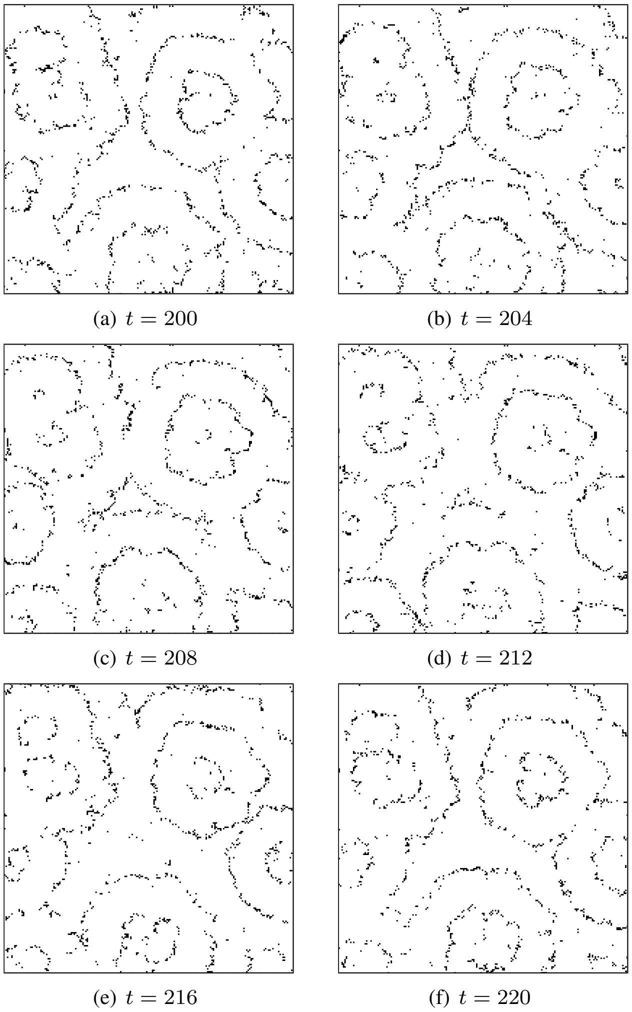
[19] T. Yan, T. He, and J. A. Stankovic. Differenciated surveillance service for sensor networks. In *Proceedings of SenSys' 03*, pages 51–62, 2003.

[20] F. Ye, G. Zhong, J. Cheng, S. Lu, and L Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS '03)*, pages 28–37, 2003.

[21] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. In *Proceedings of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, invited paper*, pages 89–124, 2004.

Appendix 1

Figure 13 gives some idea of wave motion by showing closely spaced snapshots, viz., every 4 time steps; the parameters are those of Figure 1. The emergent behavior can be seen by focusing on the seed locations and watching evolution from the wave nucleations.



(a) $t = 200$

(b) $t = 204$

(c) $t = 208$

(d) $t = 212$

(e) $t = 216$

(f) $t = 220$

Figure 13. Greenberg-Hastings Automaton on $\mathbb{R}^2$ (k=20)