International Journal of Networking and Computing – www.ijnc.org
ISSN 2185-2839 (print) ISSN 2185-2847 (online)
Volume 3, Number 1, pages 170–180, January 2013

A Transferring Protocol using Local Information on Ad Hoc Sensor Network and Its Behaviors

Dansaku Nakayama
Graduate School of Engineering
Tokyo Denki University
10kmc24@ms.dendai.ac.jp

Naoshi Sakamoto
School of Engineering
Tokyo Denki University
sakamoto@c.dendai.ac.jp

Katsunobu Imai
Graduate School of Engineering
Hiroshima University
imai@hiroshima-u.ac.jp

Received: February 15, 2012
Revised: May 29, 2012
Revised: August 31, 2012
Accepted: October 4, 2012
Communicated by Yasuaki Ito

Abstract

To communicate over an ad hoc sensor network, many routing protocols usually collect information from the whole of the network. Thus, for transmitting a message, they consume an amount of power that is proportional to the size of the network.

Now, we restrict the problem so that messages can be transferred only in a certain direction on a two-dimensional surface. If we solve this problem by employing a protocol using only local information, the protocol would consume an amount of power that is proportional to not the size of the network, but the length of the transmission path since only the nodes near the path would consume power.

Inspired by the glider of the Game of Life cellular automaton, we propose a protocol designed to obtain information on the shape and the direction of movement of each group of nodes by using only local information. Our protocol has limited straightness for a random distribution of arrangements of nodes.

Keywords: Sensor networks, Cellular arrays and automata, Routing protocols

1 Introduction

Many studies have been conducted on power consumption during communication in sensor networks. In these studies, many assumptions have been made. One study investigated the use of protocols designed for ordinal ad hoc wireless networks, while another developed a protocol that requires an additional magnetometry device[4].

In this study, instead of considering additional hardwares, we assume that many sensor nodes are distributed in an area. We propose a communication protocol with low power consumption.

Unlike ad hoc networks of PCs, one-to-one communication is not important in ad hoc sensor networks. A sensor node is equipped with some sensors, and the information gathered by the sensors is expected to be transferred to nodes, called sink nodes, that are used solely for recording information; the sink nodes are placed at the outer edge of the sensor network. Note that a sensor network can have more than one sink node. Some studies assume a group of sink nodes arranged in a row, called a wall[4]. When such an assumption is made in the case of a densely distributed sensor network, since it is not important to decide a route from a sensor node to a sink node, strict routing is not necessary to obtain information, such as the air temperature in an area. Moreover, the information may be transferred to the outer edge of the sensor network.

Now, let us consider an intuitive way to transfer signals. Let signals be transmitted along straight paths from two points on the outer edge of a sensor network and let the paths be such that they intersect at some point. Then, signals sent from both points can be obtained at the point of intersection. This might be achieved if we can implement a specific structure going straight on, like the glider in the Life Games of cellular automaton in sensor networks.

In this paper, we propose a protocol that transfers information linearly by using only local information in a sensor network consisting of sensor nodes distributed on a two-dimensional surface. We investigate the validity of this protocol and the influence of the density of nodes on the protocol through simulation and mathematical analysis. Additional studies on the protocol are in progress.

Now, we consider power consumption of protocols in a scenario where many nodes are distributed in an area over a two-dimensional surface. Theoretically, if a protocol that looks up some nodes causes traffic to all nodes, the amount of power consumption of such a protocol must be $O(N)$, where N denotes the number of whole nodes in the network. On the other hand, suppose that a protocol can transfer information by making only nodes near the transmission path work. Then, the amount of power consumption of this protocol would be $O(\sqrt{N})$ since the maximum number of working nodes is the ratio of the length of one edge to the area of the surface. Therefore, the power consumption of our protocol is lower than previously proposed protocols.

The rest of the paper is organized as follows. In Section 2, we present the notation used. We discuss related work in Section 3. The proposed protocol is presented and analyzed in Sections 4 and 5, respectively. Observations of some experiments on our protocol are shown in Section 6. Finally, we discuss future works in Section 7.

2 Preliminary

In this study, a sensor node has a size of zero and no identity number. Let the radius of the communication range of each node be one. We assume that a node consumes the power only when it sends messages. Let Ω be the set of whole nodes. Let $\mathcal{N}$ denote the set of natural numbers. Further, let $\mathcal{R}$ denote the set of real numbers.

We denote a sensor node by a lowercase letter such as a, and a set of nodes by an uppercase letter such as A. Let $[a]$ denote the set of sensor nodes that are placed within the communication area of a. In order to expand this representation for the set of sensor nodes, we denote $[A]$ as $\bigcup_{a\in A}[a]$.

Sensor nodes are distributed on a two-dimensional surface. We employ xy rectangular coordinates. Then, we present a protocol that can make patterns move in the x-axes direction. These patterns are received by information collection nodes that are placed on the line $x = g$, where g is a sufficiently large number. The information collection nodes are not only placed at $(g, 0)$, but also in the range from (g, y_g) to $(g, -y_g)$, where y_g is also a sufficiently large number. This implies that even if patterns deviate from the x-axis direction, the information can be received by the information collection nodes. By adding a payload of finite length containing information to messages to form patterns, we can easily transfer information. In this study, we focus on an algorithm and conditions required for such moving patterns. Moreover, we do not investigate how to transfer information.

Now, let us consider the property our protocol satisfies. Assume that every node that executes protocol Π takes several status at each instant. These status must include the status in which the node does not consume power and one in which the node consumes power since it sends messages. Let $S(t) \subseteq \Omega$ be the set of nodes that are consuming power at time t. Furthermore, for point $p \in \mathcal{R}^2$ and $r > 0$, let $R(p, r) \subseteq \Omega$ be the set of nodes contained in a circle with radius r and center at p. Moreover, for points p_1 and p_2, let $d(p_1, p_2)$ denote the distance between p_1 and p_2. Then, we define the"autonomous migrating protocol" as follows:

Definition 1 *Protocol Π is an autonomous migrating protocol if it satisfies the following two conditions:*

(locality)

$$(\forall t \in \mathcal{R})(\exists p \in \mathcal{R}^2)[S(t) \subseteq R(p, r)]$$

and

(migration)

$$\begin{aligned}&(\forall t_0 \in \mathcal{R})(\forall D > 0)(\exists p_0, p_1 \in \mathcal{R}^2)(\exists t_1 > t_0)\\ &\quad [S(t_0) \subseteq R(p_0, r) \wedge S(t_1) \subseteq R(p_1, r), \wedge d(p_0, p_1) > D].\end{aligned}$$

When we design protocols, it is convenient to assume that all nodes are synchronized. However, since our study assumes that many nodes are distributed densely over a wide area, it is not realistic to make this assumption. Nevertheless, protocols that make patterns go straight on, such as gliders in the Game of Life of cellular automaton, require only nodes near the route to be synchronized. We first improve a protocol that simulates a synchronized network on an asynchronous network without any special assumption such as a leader (for example Synchronizer α in [2]) in order to work in a limited area by limiting the number of times each message is transferred. Then, by employing the improved synchronizing protocol, protocols that make patterns go straight on for synchronized networks to work on asynchronous networks.

We assume that even if many nodes send messages to a node in the same round, the messages do not interfere with each other, and the receiving node can receive all messages separately. Moreover, we assume that our protocol can also absorb communication delays so that every node can receive and handle all messages sent in the same round. Each node must receive messages that are sent within its communication area, should not be capable of receiving messages that are sent outside its communication area, and should be capable of deciding whether it wants to receive a message. According to these assumptions, our protocol can use the condition of whether a node receives a message at a certain time. These assumptions can be realized with high probability by guaranteeing enough interval of each round with respect to the number of nodes and the number of messages, and by employing a lower layer protocol resending messages when it detects a collision.

3 Related Work

Under a scenario in which thousands of disposable sensors are scattered over a forest, Chatzigiannakis and Nikoletseas proposed a protocol in which once a sensor node detects an event, the sensor network transmits the information to a wall that consists of sink nodes[4]. However, their scenario is different from ours in that they assumed that each node has a sensor that can obtain the direction to the wall, such as a magnetometry device.

On the other hand, Dil and Havinga proposed a protocol for a sensor network in which beacons that know their location and nodes that do not know their location are distributed in a mixed manner[5]. The protocol makes nodes estimate their location according to the relationship between the number of hops from themselves to beacons, and compute the routing information. This protocol is similar to ours with regard to the computation of the routing information by using only local information.

Karp presented a communication protocol that searches neighbor nodes for the node nearest to the destination; the nodes are assumed to be distributed on a two-dimensional surface[3]. His assumption that nodes communicate with only neighbor nodes is similar to our assumption. However, his protocol assumes that each node knows its global position and identity.

While we were designing our protocol, we were inspired by special patterns called "gliders" in the Game of Life cellular automaton[7]. The Game of Life is the most famous rule of two-dimensional cellular automaton. The rule is that a cell is alive if there exist three neighbors; otherwise it dies. Under this rule, there exist patterns that repeat periodically, and then change their position. In this study, we design a protocol to form a repeating moving pattern. It is also possible to extend the Game of Life by increasing its neighborhood size. There are families of glider-like moving clusters of live cells with the extended rule (Larger than Life cellular automata)[6].

Further, there are studies on sensor networks employing the idea of cellular automata [1, 8, 9, 10]. However, we have not found any studies that make use of migration of patterns on stable nodes, resembling a glider, to control routing for messages without using special devices.

4 Our Protocol

We propose the following protocol: when suitable sets of nodes C_0, D_0, E_0, C_1, D_1, and E_1 are given as the initial nodes, the protocol makes nodes compute C_n, D_n, and E_n inductively for $n \geq 2$.

Protocol 1 *For $n \geq 2$, each pattern is defined as follows:*

$$C_n = [D_{n-1}] \cap [E_{n-1}] - C_{n-1} - D_{n-1} - E_{n-1} - C_{n-2} - D_{n-2} - E_{n-2}, \quad (1)$$

$$D_n = [C_n] \cap [D_{n-1}] - C_n - C_{n-1} - D_{n-1} - C_{n-2} - D_{n-2}, \text{ and} \quad (2)$$

$$E_n = [C_n] \cap [E_{n-1}] - C_n - C_{n-1} - E_{n-1} - C_{n-2} - E_{n-2}, \quad (3)$$

where the symbol $-$ denotes subtraction for sets.

Each node computes our protocol as follows:

1. In order to distinguish among sets that are adjacent to each other, the number of the index of each set is represented as a finite bit series of sufficient length, and only the lower bits are calculated.

2. The names of C, D, and E are also coded. Thus, each node can code the name of the set that it belongs to. We call the coded name of a set "local id".

3. Each node broadcasts its local id.

4. After nodes receive local ids from others, each of them calculates and determines which set it belongs to on the basis of the protocol definition.

Note that when the sets are decided up to $n-1$, only C_n is decided at first: D_n and E_n are decided in next step.

From the above discussion, we can see that this protocol requires each node to use memory corresponding to a finite number of bits.

We simulate this protocol on nodes distributed randomly. We show the result in Figure 1. We can see the series of patterns go partly straight. However, sometimes this protocol happens to stop at a random distribution of nodes.

5 Stability

In this section, we discuss the stability of Protocol 1.

First, we consider the following distribution of nodes:

Condition 1 *Nodes are distributed continuously.*

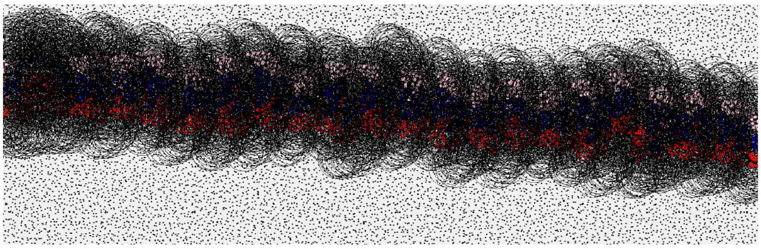

Figure 1: A results of the simulation

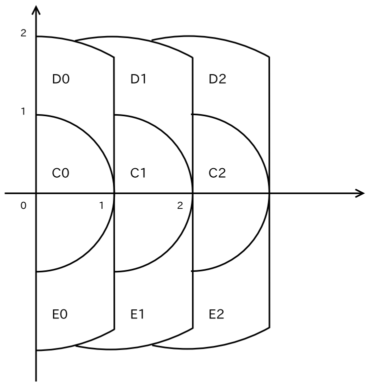

Figure 2: Stable pattern

Theorem 5.1 *For Condition 1, there exists an initial pattern to obtain congruence patterns repeatedly. That is, there exists an initial pattern that facilitates the infinite repetition of the protocol.*

Note that Condition 1 is not realistic. If a node receives infinitely many messages, it cannot handle them in a single step. However, we ignore the limitation of the computation power of each node in this theorem.

Proof.) We give the initial pattern as C_0, D_0, E_0, C_1, D_1, and E_1 [1] (Figure 2). We then show that our protocol yields pattern C_2, D_2, and E_2 that are congruent with C_1, D_1, and E_1, respectively, along the x-axis. This shows that the protocol can repeat infinitely by computing inductively.

First, we define the initial pattern as follows. Let C_0 be the right-side semicircle of the central coordinate of $(0, 0)$ with a radius of one. Let D_0 and E_0 be the areas surrounded by the right-side semicircle of the central coordinate of $(0, 0)$ with a radius of two, by C_0, and by the line $x = 1$. Let C_1 be the right-side semicircle of the central coordinate of $(1, 0)$ with a radius of one. Let D_1 be the area surrounded by the right-side semicircle of the central coordinate of $(1, 0)$ with a radius of two, the left side semicircle of the central coordinate of $(1, 1)$ and the radius of one, outer boundaries of D_0 and C_1, and the line $x = 2$. We define E_1 similar to D_1.

Now, we explain how to obtain C_2. Recall that C_2 is contained by the common area of $[D_1]$ and $[E_1]$. According to Condition 1, there exist nodes d and e of D and E near $(2, 0)$, respectively. We can assume that d and e can be selected to be arbitrarily close to each other. Thus, both the communication areas d and e almost coincide with a circle of the central coordinate of $(2, 0)$ with a radius of one. Let C_2 be the area generated by subtracting C_1, D_1, and E_1 from this circle. This gives the right-side semicircle. Thus, C_2 is congruent with C_1.

Next, we consider D_2. As described in Figure 3, let us consider the common area of $[C_2]$ and $[D_1]$. First, we consider the outer boundary of $[C_2]$. Recall that $[C_2]$ denotes an area enclosed by circles whose centers are contained in C_2 and whose radii are equal to one. $[C_2]$ can be divided

[1] These patterns are obtained by trial and error.

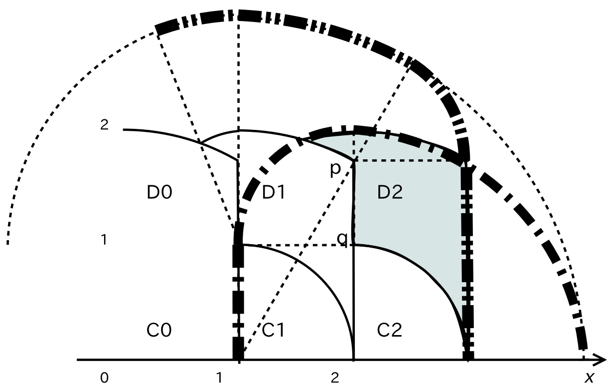

Figure 3: Outer boundary of $[D_1]$ and $[C_2]$

into three parts. The communication area of the nodes around the arc of C_2 corresponds to the right-side semicircle of the central coordinate of $(2, 0)$ with a radius of two. On the other hand, the nodes around the code can communicate with nodes up to the line $x = 1$. Now, we focus on a node around the point of intersection $(2, 1)$ (say q) of the arc and the chord. Its communication area forms a circle of the central point of q with a radius of one. We draw the outer boundary of $[C_2]$ as a dot-dash line in Figure 3.

Second, we consider the outer boundary of $[D_1]$. We focus on the common part between $[D_1]$ and $[C_2]$. The communication area of the nodes around the line $x = 2$ that surrounds D_1 reaches the line $x = 3$. On the other hand, the communication area of the nodes around the arc of the central coordinate of $(1, 0)$ with a radius of two that surrounds D_1 is bounded by the arc of the central coordinate of $(1, 0)$ with a radius of three. Now, we focus on the point of intersection(say p) of the arc of D_1 and the line $x = 2$. The communication area of a node around p corresponds to the arc of the central coordinate of p with a radius of one. Since the x-coordinate of p is equal to 2 and since the distance between p and $(1, 0)$ is equal to two, the coordinate of p is $(2, \sqrt{2})$. Therefore, this arc connects to both the line $x = 3$ and the arc of the central coordinate of $(1, 0)$ with a radius of three. We draw the outer boundary of $[D_1]$ as a dot-dot line in Figure 3.

Finally, we find that one of the points of intersection of the boundaries of $[C_2]$ and $[D_1]$ is $(3, \sqrt{2})$. Then, we can easily verify that D_2 is congruent with D_1.

It can similarly be shown that E_2 is congruent with E_1. □

Next, we consider the following sparse distribution.

Condition 2 *There exists at least one node in any circle of radius r.*

We consider the condition for r to repeat our protocol.

Theorem 5.2 *In Figure 2, for C_0, D_0, E_0, C_1, D_1, and E_1, let each of C_0, D_0, E_0, and C_1 satisfy Condition 1 while the remaining area containing D_1 and E_1 satisfy Condition 2. Then, if $r < \frac{1}{8}$, C_2 must have a node.*

Proof.) The outer boundary of C_2 is decided by the communication areas of the node of D_1 and the node of E_1 that are close to each other. Thus, we look for the condition of r such that for the closest pair of node d (in D_1) and node e (in E_1), their common communication area can contain a circle of radius r. Let s_d be the inscribed circle of radius r in D_1 near E_1. We define s_e in E_1 similarly. The length h of the perpendicular line from the center of s_d to the x-axis is as follows:

$$(1 - r)^2 + h^2 = (1 + r)^2. \tag{4}$$

On the other hand, since the outer boundary of C_2 is determined by a node in s_d and a node in s_e, we select a node d from s_d and a node e from s_e in order to minimize C_2 (Figure 5.2). In order that

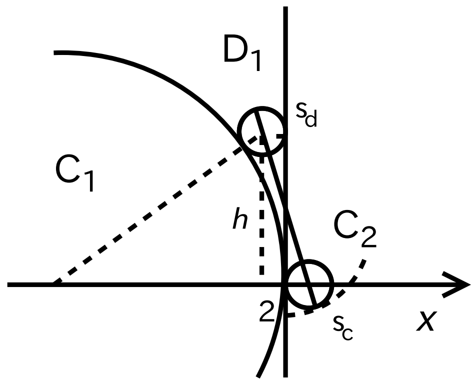

Figure 4: Relationship between the inscribed circles

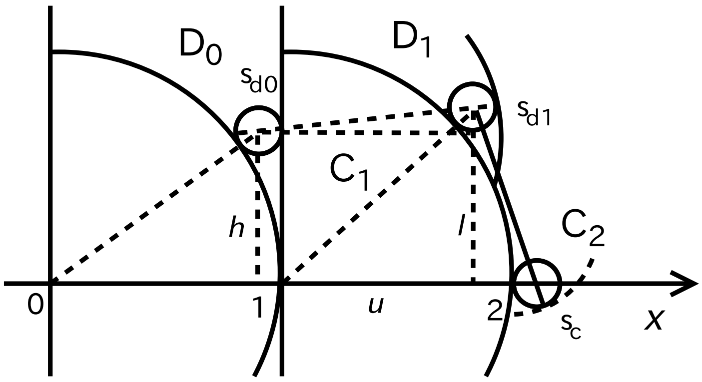

Figure 5: Relationship among the inscribed circles

C_2 contains a circle of radius r irrespective of the selected d and e, the distance between the centers of s_d and s_e must be smaller than $1-2r$. Then, we have the following inequality:

$$(2r)^2 + h^2 < (1-2r)^2. \tag{5}$$

From (4) and (5), we obtain $r < \frac{1}{8}$. □

The larger the distance between the selected nodes of D_{n-1} and E_{n-1}, the smaller is the size of C_n. Thus, for the distance to be large, C_{n-1} should be larger. Next, we consider the condition that C_2 has a node when the size of C_1, which is the right-side semicircle, is the maximum.

Theorem 5.3 *Suppose that Condition 2 is satisfied everywhere. In Figure 5, our protocol is given C_0, D_0, E_0, and C_1 as the initial set of nodes. Then, if $r < 0.076$, C_2 must have a node.*

Proof.) Similar to the case of Theorem 5.2, according to Figure 5, we introduce the condition by showing the relationship among the conditions of existence for nodes. Consider a circle s_{d0} with radius r in the area D_0. According to Condition 2, there must a node in this circle. Further, consider a circle s_{d1} with radius r in the area D_1 and a circle s_C with radius r in the area C_2. By considering the symmetry, we assume that the center of s_C is located on the x- axis.

This theorem can be proved by obtaining the condition for s_C to be in the area C_2.

Let h be the length of the perpendicular line from the center of s_{d0} to the x-axis. Similarly, let l be the length of the perpendicular line from the center of s_{d1} to the x-axis. Moreover, let the distance between the point of intersection of the perpendicular line and the x-axis, and point $(1,0)$ be u.

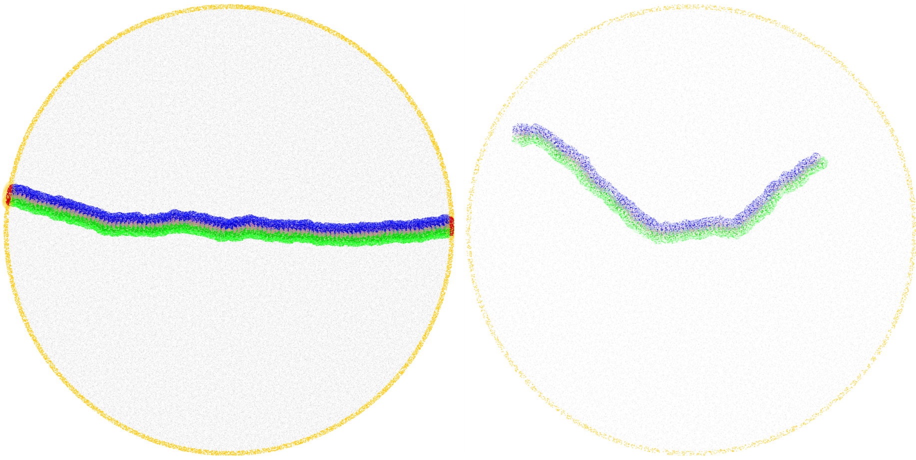

Figure 6: Random Distribution

In order that C_2 contains s_C, the communication area of a node in s_{d1} must contain s_C. Therefore, the following inequality holds:

$$(1 + r - u)^2 + l^2 < (1 - 2r)^2. \quad (6)$$

By letting s_{d0} be in contact with both C_0 and C_1, we find that the node nearest to the x-axis must exist at least in s_{d0}. Then, we have the following condition for s_{d0}:

$$h^2 + (1 - r)^2 = (1 + r)^2. \quad (7)$$

Similarly, to bring s_{d1} close to the x-axis, we have the following conditions:

$$\begin{aligned} (l - h)^2 + (r + u)^2 &= (1 - 2r)^2, \text{ and} && (8) \\ u^2 + l^2 &= (1 + r)^2. && (9) \end{aligned}$$

From the above formulas, we have the following inequality:

$$-4r^3 + 24r^2 + 51r - 4 < 0. \quad (10)$$

Then, by numerical analysis, we obtain the following condition:

$$-1.725 < r < 0.076, r > 7.650.$$

Since r must be contained in $(0, 1]$, we obtain the condition $0 < r < 0.076$. □

6 Experimental Observation

In this section, we show some behaviors of our protocol for certain distributions of nodes.

We design a simulator using C# language. Our simulator processes each node every round according to our assumptions. It does not simulate the propagation of radio waves. It only guarantees that each message must reach nodes within the communication area of its sender, and must be processed in the same round. It displays the movement of the information by changing the color of nodes with respect to their state by their communication.

6.1 Random Distribution

In this paper, we have not discussed how our protocol should be started. On the other hand, we assume that there exists a receiving wall and the pattern starts moving toward the wall. For this

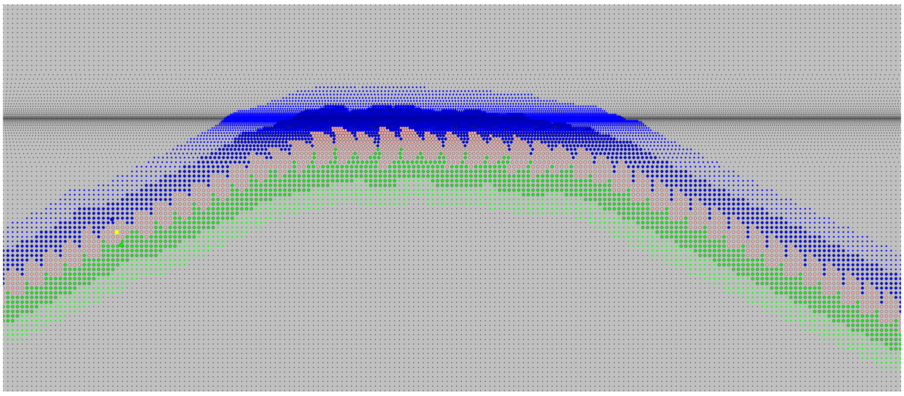

Figure 7: Refraction

assumption to hold true, nodes to start our protocol must have information on the direction of the wall. Thus, additional assumptions are required.

Accordingly, we made the receiving wall enclose a region. For example, in order to receive information about animals in the woods, we would enclose the woods with the receiving wall. In this case, no matter which direction the protocol starts with, patterns can reach efficiently when the protocol has straightness. Moreover, we do not have to add any assumption, except for the way to position of the receiving wall. On the other hand, we don't focus on how to manage the receiving wall efficiently. Nevertheless, even if the receiving wall consists of other kind of sensor nodes, the number of required nodes is at most $O(\sqrt{N})$. Moreover, for sensor network that consists of small nodes like smart dusts, the assumption that we make the receiving wall enclose the region might not be very unrealistic.

In Figure 6, we show the results of our simulation; in the simulation, the receiving wall is set to be in the form of a circle with radius 50, nodes are distributed randomly, and the protocol starts at the center of the circle. In this simulation, we make the protocol start by giving condition D to some nodes in an area and condition E to some nodes in another area with the two areas being in contact. Such a start yields two patterns moving toward opposite sides. In the left image, the protocol reaches the wall while maintaining straightness for a dense distribution. On the other hand, in the right image, it bends and it stops for a sparse distribution. Each distribution of these results (for $r = 0.1$, 0.2) is obtained as follows:

1. We achieve triangulation by using regular triangles with side length r over the whole area.
2. We put a node on each vertex of all triangles. Thus, Condition 2 is satisfied.
3. We move every node over a uniform random distance, from zero to one.

With these steps, the distribution does not satisfy Condition 2 strictly, but the average density may approximate the distribution of Condition 2.

Therefore, we find that these results of our simulation support our proposition that the straightness of our protocol depends on the density of the distribution of nodes.

The condition of this simulation is limited. However, the result of simulations indicates that our protocol works well in conditions less stringent than those specified by the theorems.

6.2 Refraction

In this section, we show how the direction of migration of the pattern can be controlled by changing the density of nodes. In Figure 7, the position $p_{i,j}$ of each node $a_{i,j}$ (i and j are integers) is given as follows:

$$p_{i,j} \quad = \quad ((2i+1)k(j), \mathrm{sign}(j)sk(j)) \in [0, 30] \times [-3, 6]$$

where

$$\begin{aligned} k(j) &= \begin{cases} 0.01 + 0.0025|j| & \text{if } |j| < 16 \\ 0.05 & \text{otherwise, and} \end{cases} \\ sk(j) &= \sum_{m=0}^{j} 2k(m) - k(j) - k(0). \end{aligned}$$

In Figure 7, r becomes the minimum value of about 0.01473 in the vicinity of $y = 0$. On the other hand, r becomes $0.1\sqrt{2}$ in the region of $|y| > 1.055$ since nodes are located there in the form of grid at intervals of 0.1. We put an initial pattern around of point (3,3) in order to go into the line $y = 0$ with angle of incidence of 70 degree. Then, we have a dense area near the row $a_{i,0}$ ($i \in \mathcal{N}$). Now, we make the pattern move in an aslant direction relative to the dense area. The direction of the pattern is bent like refraction. This property is opposed to the property of light. Nevertheless, by putting two dense rows in parallel, we can send a message along the parallel dense rows like an optical fiber.

7 Future Work

In this paper, we proposed a transferring protocol using local information for a sensor network and analyzed the conditions to be satisfied for the execution of the protocol. However, we have not obtained the condition to execute the protocol repeatedly yet. We are studying this condition as well the possibility of using the protocol for a randomly distributed sensor network. Moreover, the average velocity and the straightness should be investigated for any random distribution.

References

[1] Dylan G. Allegretti, Garrett T. Kenyon, and William C. Priedhorsky. Cellular automata for distributed sensor networks. *International Journal of High Performance Computing Applications*, 22(2):167–176, May 2008.

[2] B. Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32(4):804–823, October 1985.

[3] Karp Brad. GPSR: Greedy perimeter stateless routing for wireless networks. *Proc. MobiCom*, 8 2000.

[4] Ioannis Chatzigiannakis and Sotiris Nikoletseas. A sleep-awake protocol for information propagation in smart dust networks. In *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, IPDPS 2003, page 225, 2003.

[5] B.J. Dil and P.J.M. Havinga. COM-LOC: A distributed range-free localization algorithm in wireless networks. In *Proceedings of the 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2009*, pages 457–462, 12 2009.

[6] Kellie Michele Evans. Larger than life: Digital creatures in a family of two-dimensional cellular automata. In *Discrete Mathematics and Theoretical Computer Science Proceedings, Vol. AA (Discrete Models: Combinatorics, Computation, and Geometry)*, pages 177–192, 2001.

[7] Martin Gardner. Mathematical games, the fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, 223:120–123, October 1970.

[8] Maryam Kalantary and Mohammad Reza Meybodi. Energy-aware routing protocol for mobile sensor networks using learning automata algorithms. In *WiMob*, pages 492–496, 2010.

[9] R Manoharan, P Ajit Jose Sylvester, and M Ramprasath. A robust routing protocol for vanets using cellular automata. *International Journal on Recent Trends in Engineering and Technology*, 3(2):141–143, 2010.

[10] A H F Navid and H H S Javadi. ICLEAR: Energy aware routing protocol for wsn using irregular cellular learning automata. *Industrial Electronics*, 1(Isiea):463–468, 2009.

[11] Andrew Trevorrow, Tom Rokicki, Tim Hutton, Dave Greene, Jason Summers, Maks Verver, and Robert Munafo. Golly game of life home page. http://golly.sourceforge.net/.