EAI Endorsed Transactions
on Context-aware Systems and Applications
Research Article ICST.ORG

Synchronous networks for bio-environmental surveillance based on cellular automata

Bao Hoai Lam[1,*], Hiep Xuan Huynh[2], Bernard Pottier[1]

[1]Université de Bretagne Occidentale, LAB-STICC, UMR CNRS 6285, France

[2]DREAM Team/UMI 209 UMMISCO-IRD, Cantho University, Vietnam

Abstract

The paper proposes a new approach to model a bio-environmental surveillance network as synchronous network systems, systems consist of components running simultaneously. In the network, bio-environmental factors compose a physical system of which executions proceed concurrently in synchronous rounds. This system is synchronized with a synchronous wireless sensor network, the observation network. Topology of the surveillance network is based on cellular automata to depict its concurrent characteristic. Several aspects of the above model is simulated by using the case study *Brown Planthoppers surveillance network in the Mekong Delta*.

Received on 06 January, 2016; accepted on 28 January, 2016; published on 09 March, 2016

Keywords: Bio-environmental surveillance network, Brown Planthopper, synchronous network, cellular automata.

Copyright © 2016 B. H. Lam *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution license (`http://creativecommons.org/licenses/by/3.0/`), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eai.9-3-2016.151117

1. Introduction

Bio-environmental phenomena occur continuously and concurrently. Continuous occurrence means that they compose an unbroken whole, without interruption while concurrency allows them to happen at the same time. For example, some factors such as temperature and wind influencing Brown Planthoppers (BPHs) invasion [6][7] from one place to another, are continual. Besides, they are concurrent since the motivation for propagating from a source to a destination comes from surrounding conditions of the source and its neighbors. Conditions from such different places must be executed simultaneously. Similarly, floods which cause overflows of water are also uninterrupted and concurrent.

To confront with disasters caused by bio-environmental phenomena, several experienced as well as academic solutions are proposed and a solution with wireless sensor network (WSN) emerges as a suitable capable choice [4][8]. This kind of solution uses sensors to measure environments and other factors. The sensed values from sensors will be sent via a wireless network to a data center periodically. Next, a back-end system will manipulate these values and propose solutions relating to collected situations of surrounding conditions. Such application is called bio-environmental surveillance network.

There exists a relation between a bio-environmental phenomenon and its observation WSN in a surveillance network. Indeed, the WSN provides sampling inputs for the phenomenon and the phenomenon occurs in accordance with its rules based on these inputs. On the other hand, damage levels of bioenvironments caused by the phenomenon are sampling measured by sensor nodes of the WSN.

The paper proposes a new approach to model a bio-environmental surveillance network as synchronous networks [14] in order to emerge the relation between bio-environmental phenomena and their WSNs. In this work, the topology of the surveillance network is based on cellular automata [24], a parallel structure.

The structure of this paper is as follows. Section 2 depicts some previous work relating to wireless sensor network as well as bio-environmental surveillance modeling. Next section is about modeling a bio-environmental surveillance network as synchronous networks. Definitions of a synchronous network and a synchronous wireless sensor network are also depicted in this section. To be an example for this model, case study Brown Planthopper surveillance network in the Mekong Delta of Vietnam is introduced in section 4. A data model for a bio-environmental surveillance network is described in section 5. Section 6 depicts

---

[*] Bao Hoai Lam. Email: lhbao@cit.edu.vn

implementations of a bio-environmental surveillance network. Next section illustrates some simulation results of the surveillance network described in the case study. The last section is our conclusion and future plans.

2. Related work

Examples of bio-environmental surveillance network is variant such as light trap networks, flood surveillance, fire forest surveillance. Light trap network uses light traps to measure environmental factors and insect densities. Thanks to it, people can make suitable decisions to get better protection for their crops from insect attacks.

Light trap method is one of solutions to prevent high densities of spruce budworm in Canadian forests [9][10]. It allows people to participate insect trapping by giving light traps to them and track their traps from June to end of August every year. Periodically, people only report estimated densities of insects via a website, an application or even with a paper and a pen. Finally, trap samples are collected and counted in a lab environment. Applications of data collections from these light traps are variant, for example, thanks to wing wear and body size measurements of adult spruce budworms captured at light traps in some previous years, some useful inference on seasonal patterns related to reproduction can be archived [11]. However, these light traps seem not to compose a network, instead, they create a combination of traps to collect data for post processing. Therefore, there are few information about the model of the light trap network.

A forest fire detection system can be modeled as a k-coverage problem in WSNs [36]. In this work, Fire Weather Index (FWI) System [37] is used in designing an efficient fire detection system in order to optimize the communication and sensing modules of the observation WSN. The WSN life circle is prolonged due to a data aggregation schema based on the FWI since the schema only delivers the data that is of interested by the application.

An insect surveillance network is modeled as an interaction between an insect physical system and a WSN system, both can be modeled as synchronous networks [1]. In this model, the physical system is considered as a synchronous network based on cellular automata topology by being dividing into cellular cells. This physical system is synchronized with a synchronous WSN system. Nevertheless, data communication aspect of the WSN system seems not to be focused in this work.

3. Bio-environmental surveillance network

3.1. Synchronous network

Synchronous network [14] is a network describing synchronized rounds of message exchange and computation. It consists of pieces of processes which may send and receive messages simultaneously.

Mathematically, a synchronous network can be considered as a graph G where processes are located at its nodes and these processes communicate together via their edges using message sending.

Each node in a synchronous network is termed as a process which consists of the following components:

- $states_i$: a collection of states at process i.
- $msgs_i$: a message-generation function specifies that the process i sends to an indicated neighbor, starting from the given state.
- $trans_i$: a state-transition function specifies a new state to which the process i moves from the current state and messages from incomming neighbors.

Wireless Sensor Network A Wireless Sensor Network (WSN) [17] consists of n wireless sensor nodes distributed in a two dimensional planes (figure 1). It can be considered as a graph G=(V,E) where each sensor node is a node in V and an edge of E is established between 2 nodes if the distance between them is at most a transmission range r_t, the maximum distance that the single transmission of a node can be received by all nodes in its vicinity.

Besides, each sensor node can measure its surroundings within a sensing range r_s. Normally, the sensing range of a node is much smaller than the communication range.

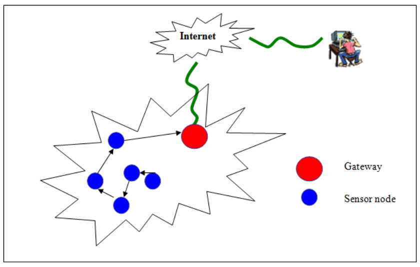


Figure 1. Sensor nodes in WSN.

Gateway node is a special node in which data sensed from other sensor nodes is integrated. Commonly, there is an application connecting to this node to process these pieces of data for decision making.

A WSN can be considered as a synchronous network [19] since the WSN shares a reference clock and allows sensor nodes periodically listening and emitting interleaved with silences. The WSN behaviors also require time division, frequency or channels division in operating the data exchange among sensor nodes.

In practice, both message-generation and state-transition functions can be shortly called as "transition rules", rules allowing the process i to send messages to neighbors in order to compose its new state.

3.2. Description of bio-environmental surveillance network

A bio-environmental surveillance network is a network to monitor bio-environmental phenomena due to environmental factors based on WSN approach [4][5][8]. In this network, there are 2 distinct systems: a physical system and a network system which interact and exchange data together in order to compose a whole one: a *bio-environmental surveillance network* (figure 2).

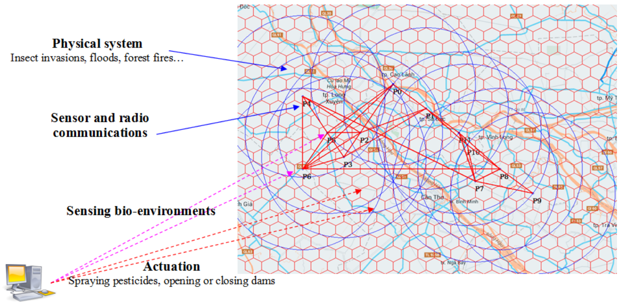


Figure 2. A bio-environmental surveillance network is composed of 2 systems. A physical system is a phenomena' working space which is divided as cells. These cells are monitored by a WSN system (*sensor and radio communications*).

A physical system is a system that the phenomena occur, in other words, it is a working space or working environment of the phenomena. This space consists of environmental factors and other conditions. For example, brown planthoppers desire to suck rice plants under some environmental factors, therefore, rice fields and these factors compose a system termed: *physical system*.

The above physical system is monitored by computing components which connect together to constitute a wireless network. It is a WSN in which sensor nodes can periodically sense factors in the physical system. Next, sensed values are transmitted through the wireless network to a data center for post processing.

In short, the integration of the physical system and wireless network system composes the bio-environmental surveillance network. Both systems can be modeled as synchronous networks.

3.3. Physical system

Description. The *physical system can be considered as a synchronous network* since components of this network can be found in the physical system.

The physical system of a bio-environmental surveillance network is the working space (or environment) in which the phenomena occur. This space is divided as units called *cells* (figure 2). Each cell has 4 neighbors (Von Neumann neighborhoods), 8 neighbors (Moore neighborhood), or 6 neighbors (hexagonal cell) [25] (figure 3).

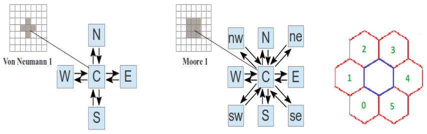


Figure 3. Neighbors of a cell.

The space in figure 2 represents a cellular automaton which is depicted by a triple (S, n, f) where:

1. A finite state set S. A state of a cell describes the status of that cell.
2. Distance n identifying neighbor cells, *normally n=1*. When *n=1*, a cell has at most 4, 6, or 8 surrounding cells.
3. Transition rule f: $S^n \rightarrow S$ depicts the change of a cell's state at a specific time based on the current state of the cell and its neighbors.

   At the time t, the state of a cell depends on the state at time t - 1 of its neighbors. The cell itself can be integrated in its neighborhoods. Updating cells are done by a transition rule. All cells have the same transition rule and the transition rule is applied to all cells at the same time. Whenever the rule are applied to the entire system, they could change the entire system synchronously.

Mathematically, the above cellular automaton of the physical system is a topology of **a synchronous network** which is modeled as a directed graph $\mathbf{G_1}=(V_1, E_1)$ where V_1 is the collection of nodes and E_1 is the set of edges.

Nodes. Each cell in figure 2 represents a node in the graph $\mathbf{G_1}$. By the time passing, each node i (also called process i) holds a collection of states $states_i$ of which a state describes a status of the cell at a time t.

Edges. Edges in the graph $\mathbf{G_1}$ are composed by links between a node and its neighbors (4, 6, 8 neighbors in figure 3). Because $\mathbf{G_1}$ is a directed graph, there are 2 edges between the node and a neighbor of it (2 directions). Thus, these 2 edges can become an undirected edge and $\mathbf{G_1}$ can be seen as an undirected graph.

Behaviors. Behaviors at node i are expressed by transition rules of states $transitions_i$. Normally, these rules are functions which specify a set of conditions causing bio-environmental phenomena.

3.4. Synchronous Wireless Sensor Network

Sensor nodes are distributed in the some cells of the physical system (circles in figure 2) to measure some factors in the physical one. These sensor nodes can compose a synchronous WSN (section 3.1) in which each node is a node of a graph $\mathbf{G_2}$ and each edge between 2 nodes is identified thanks to their communication ranges.

In this case, behaviors of the WSN can be operated as followed:

1. Firstly, states for all sensor nodes are initialized to prepare output messages. These states is the initial environmental conditions and insect behaviors sensed from the physical insect system. They are also messages sent through the network.
2. Next, a loop will be executed forever throughout each node.
    - Each node sends messages to its output neighbors.
    - States of the node and neighbors are changed according to their current states and input messages
    - It prepares next output messages.

3.5. Cyber Physical System

A Cyber Physical System (CPS) [13] is a system of collaborating computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. In physical world, the passage of time and concurrency are two core characteristics.

The bio-environmental surveillance network fits into a CPS framework. Environmental factors become physical entities while the observation network is the computation. Sensor nodes in the surveillance network can sense the surrounding conditions and thanks to monitoring the data collected in sensors, people can have some decisions relating to their situations. These build a physical loop between physical entities and computation with timed characteristics.

3.6. BIOSYN

Both physical system and network system are modeled as synchronous networks, therefore, the insect surveillance network can be called as **BIO**environmental **S**urveillance s**Y**nchronous **N**etworks (BIOSYN).

It is necessary to have some synchronizations between these two systems in operating such BIOSYN. The point is that the physical system is a continuous system while the WSN is a discrete one with an interval time between two adjacent sensing times. For example, it is clear that the cricket invasion is continuous, however, the WSN, if applicable, does its jobs discretely. Thus, both physical system and network system need to have some synchronous points to synchronize the data exchange between them (figure 4).

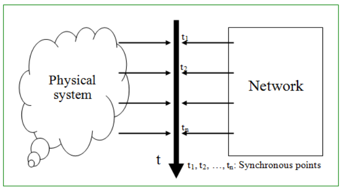


Figure 4. Synchronous points in BIOSYN.

4. Case study: The Brown Planthopper surveillance network in the Mekong Delta

4.1. Problematic

Light trap network is one of solutions for insect management in Vietnam. The network uses light traps to attract insects (due to their phototactics) in order to classify and count insect densities. Thanks to these density values, people can know situations of their fields better and make decision if necessary.

The Wireless Sensor Network approach applied to a light trap network (as proposed solutions in [4][5]), may help calculating Brown Planthopper (BPH) densities and measure environmental factors automatically. This kind of solution uses sensors, new automatic light traps, to measure environments and hopper behaviors. These sensed values from sensors will be sent via a wireless network to a data center periodically. Next, a back-end system will manipulate these values and propose solutions relating to situations of collected data. Such application is called BPH surveillance network.

4.2. Synchronous networks for BPH surveillance

A BPH surveillance network is a network to monitor BPH behaviors due to environmental factors based on WSN approach [4][5][8]. The network consists of 2 systems: insect physical system and network system [1]. These systems can be considered as synchronous networks which use cellular automata as their topologies.

In this network, working space or working environment of hoppers is divided as a grid of cells (a cellular automaton) (figure 5). Some cells of the grid contain automatic light trap sensor nodes to sampling measure surrounding conditions and hopper densities. These sensor nodes compose a massive coordinated sensing machine which its topology is similar to mesh connected WSN.

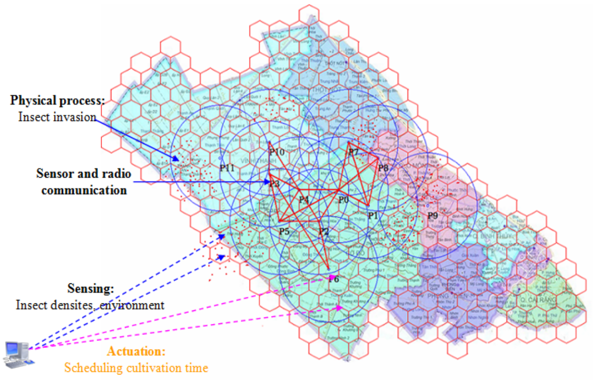


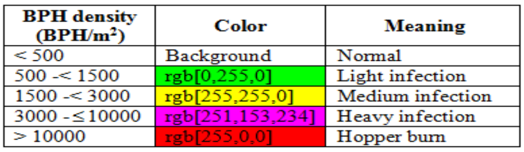
Figure 5. A BPH surveillance network is composed of 2 systems. A insect physical system is the insect's working space which is divided as cells. These cells are monitored by a WSN system.

4.3. Insect physical system

The insect physical system represents the insect working space which is divided as cells. Each cell has 4 neighbors (Von Neumann), 6 neighbors (hexagonal) or 8 neighbors (Moore).

The space in figure 5 represents a cellular automaton (**S**, **n**, **f**), where:

1. A finite state set S. A state of a cell describes the hopper status of that cell. This status is calculated thanks to the density of hoppers at that cell. In practice, people use following table [6] (figure 6) to depict hopper statuses in their fields:

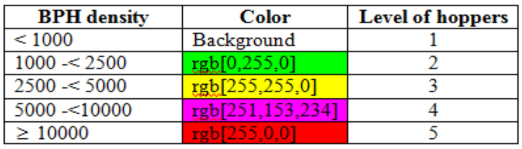


BPH density (BPH/m^2)	Color	Meaning
< 500	Background	Normal
500 -< 1500	rgb[0,255,0]	Light infection
1500 -< 3000	rgb[255,255,0]	Medium infection
3000 -≤ 10000	rgb[251,153,234]	Heavy infection
> 10000	rgb[255,0,0]	Hopper burn

Figure 6. Ascending levels of infested BPHs in rice fields

Therefore, at the time t, a cell in the hexagonal CA can be valued as an element of the set *{Normal, Light, Medium, Heavy, Burn}*.

In addition, a cell may locate an automatic light trap sensor node. This trap can catch hoppers and the density of hoppers in the trap may indicate the real infected situation at that cell. The following table [30] (figure 7) is used for describing hopper statuses in a sensor node:



BPH density	Color	Level of hoppers
< 1000	Background	1
1000 -< 2500	rgb[0,255,0]	2
2500 -< 5000	rgb[255,255,0]	3
5000 -<10000	rgb[251,153,234]	4
≥ 10000	rgb[255,0,0]	5

Figure 7. Ascending levels of infested BPHs in light traps.

2. Distance *n* identifying neighbor cells, *normally n=1*. When *n=1*, a cell has at most 4, 6, 8 surrounding cells (depends on neighbor type choice).

3. Transition rule f: $S^n \rightarrow S$ depict the change of a cell's state at a specific time based on the current state of the cell and its neighbors. For example, if the center cell and its neighbors have the state Normal at the time t, then the state of that cell at the time t+1 is Normal: f(NNNNNNN) = N.

   In fact, the transition rule f is a function depending on some variables such as: density of hoppers in a cell as well as its neighbors, rice age, wind, hopper velocity and other environmental factors.

   - Rice age: The young rice is a very good food for hoppers, therefore, they tend to locate at the young rice fields [6][7][15]. On the other hands, hoppers can not suck ripe rice so they will propagate to other fields due to wind if their rice in their current fields become mature or ripe. In addition, young rice is the first condition for hoppers landing. The green color of young rices mapped into the water is a very attractive color source for hoppers, therefore, they tend to take landing to the young ones. On the other hands, ripe rice color does not attract hoppers because they are not sensitive with this color.
   - Wind: the wind velocity, calculated in cells/time step. It illustrates the maximum distance that adult hoppers can propagate in a time step. For example, 5 cells/t means that hoppers can propagate to another cell with the distance 5 from the current cell under the wind direction. If there is no wind, hopper can transmit to its neighbor cells. Only a part of adult one can propagate to other fields. In this paper, it is an predetermined constant.
   - Hopper velocity. Without wind, hoppers can propagate to near rice fields by this their velocities, approximate 0.4m/s [16].
   - Hopper age. Totally, the life circle of BPHs is 26-30 days [15] depending on environmental factors and it spreads in 3 phases: eggs, nymphs and adults. The growth time lapse of each phase is as followed: eggs 6-8 days, nymph 12-15 days, adults 19 days. Some experiments show that a female adult BPH can lay 100-300 eggs during its life circle [15].
   - Density of hoppers in a cell and its surroundings. The relation between the

density of a cell and its state depicted in figure 6.

Mathematically, the above cellular automaton is a topology of **a synchronous network** which is modeled as a graph $\mathbf{G_1}$=(V_1, E_1) where V is the collection of nodes and E is the set of edges.

Nodes. Each cell in figure 5 represents a node in the graph $\mathbf{G_1}$=(V_1, E_1). By the time passing, each node (or process) i composes a collection of states $states_i$ of which a state holds values of rice age, wind, and BPH density at the time t.

Each node may consist a sensor node to sense above factors of a state. When the sensor node senses environment, it transmits the collected data to a gateway for storing and post processing.

Edges. Edges in the graph G_1 are composed by links between a node and its neighbors (4, 6, 8 neighbors). Because G is a directed graph, there are 2 edges between the node and a neighbor of it (2 directions).

Example. Figure 8 is an example of a graph of the BPH surveillance network in Phongdien district, Cantho, Vietnam. In this example, the map of this district is divided as hexagonal cells, each cell is almost a commune of the district. For example, if Phongdien and Nhonai communes are considered as a hexagonal center (**Center cell** in the figure), following communes such as Truonglong, Tanthoi, Giaixuan, Mykhanh, Nhannghia (hexagons 1, 2, 3, 4, 5) become neighbors of the hexagonal center approximately. The hexagon 0 is another neighbor of the center, however, it seems to occupy few area of Phongdien district.

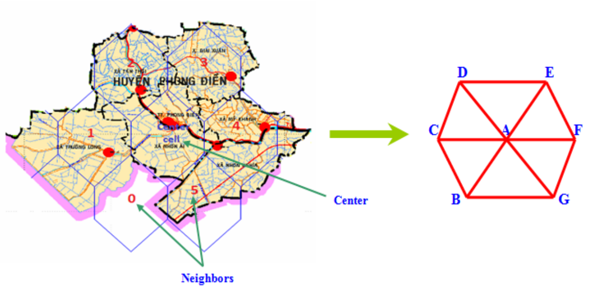


Figure 8. The graph of Phong Dien district, Cantho when it is divided as hexagonal cells.

The graph of the BPH surveillance network of this district is illustrated in the right of figure 8. In this graph, each hexagonal cell becomes a node and 2 cells in a neighborhood compose an edge. The **Center cell** is illustrated as a node A which has 6 neighbors named B, C, D, E, G, F (corresponding to cells 0, 1, 2, 3, 4, 5 respectively). However, each of B, C, D, E, G, F only has 3 neighbors.

Behaviors. Behaviors at node i are expressed by transition rules of states $transitions_i$. Normally, these rules are functions mapping a collection of states at a cell and its neighbors at the time t to create the new state of that cell at the time t+1. These transition rules are applied simultaneously at every cell.

The following pseudo code depicts the transition rule of a node n:

ALGORITHM: Transition rule (behaviors) of node n at the time t

INPUT:

d[n,v]: density of v-days old BPH at the time t-1 (v=1..hopperLifeCircle)

environment: surrounding conditions including wind, temperature, humidity, etc

OUTPUT: state of the node n

```
//At the time t, BPHs increase 1 day old
for(i=2; i<= 28; i++)
  d[n,v] = d[n,v-1];

// Egg: 1..6 days, nymph: 7-14 days;
// adult: 15-28 days
eggDays = 6;
nymphDays = 14;
adultDays=hopperLifeCircle;

//Choose hoppers in day 15-28 to give Eggs,
//if these hoppers were used before,
//choose other ones
//When eggs are created,
//they are accumulated into d[1]
layEggs(15,28,d[n], Environment);

//Calculate death rate with a rate r
calculateHopperDeath(d[n], r, environment);

//density of node n
density = calculateDensity(d);
if (density >= THRESHOLD or
    environment.riceAge IS NOT young)
 if (environment.windVelocity==0) //no wind
    for (j in neighbors of n){
      //Calculate the number of
      // migrating adults from n to j
      propagateHopperNoWind(d,n,j);
    }
    else // has wind velocity
      for (j in neighbors of n)
        if (j is in leeward of n){
          //Calculate the number of migrating
          // adults from n to j  according to
          // the wind velocity
          propagateHopperWind(d,n,j,
                    environment.windVelocity);
        }
    }
```

```
  }

  //calculate incoming hoppers from neighbors
  for(j in neighbors of n){
    calculateIncominghoppers(d,n,j);
  }

  //calculate state of n at the time t
  state = calculateState(d,n);
}
```

Hopper propagation under wind direction In the above algorithm, the method **propagateHopperWind**(*d*, *n*, *j*, *environment.windVelocity*) calculate the number of adult hoppers migrating from n to j. This hopper migration number can be calculated as followed:

Let node *n* is a source cell which is able to propagate *d adult hoppers* under the wind velocity *v (m cells/t)*. Thus, the wind velocity causes adult hoppers distribute to *at most m cells* in a period of time *t*. In this case, the number of hoppers propagating to a cell n_k which has *distance k* from the *source cell n* under the wind direction, is estimated as $\frac{d}{2^k}$ (figure 9).

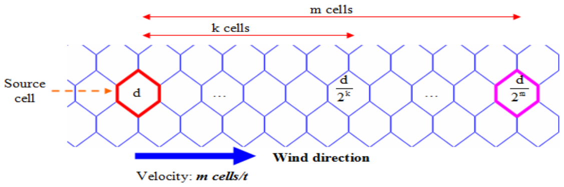


Figure 9. Estimationof propagatedadult hoppersdue to wind.

4.4. HopperobservationWireless Sensor Network

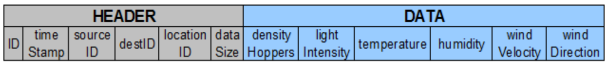
A WSN is created by distributing sensor nodes in some cells to sampling measure environmental factors and hopper density (sensor P1, P2,... in figure 5). Data collections from the WSN can be considered as indications for the operating of the whole BPH surveillance network as well as for post processing. The WSN can be considered as a synchronous network.

The WSN represents a mesh network by being considering as a graph $\mathbf{G}_2$=(V_2, E_2) where V_2 is a collection of nodes and E_2 is a collection of edges. A node in V_2 represents a sensor node while an edge is created between 2 sensor nodes if their distance is at most the transmission range (figure 5). In addition, each sensor node in the WSN has its own behaviors.

Behaviors. Behaviors of the WSN depict the data dissemination inside the network. This data dissemination includes data sensing, packaging, and transmitting via the network topology.

Sensor nodes of the WSN can sense environmental factors and hopper density. Sensor nodes are able to measure temperature, humidity, wind, light intensity, which are factors influencing hopper behaviors. Besides, nodes can also support in situ hopper density counting or estimation.

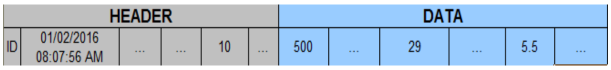
Data packet Before transmitting, data is packed as packets which each of them has the structure as in figure 10. In this structure, ID is the unique ID of the packet and location ID represents the ID of the sensor node that senses environmental factors. The time that the sensor node senses data is described by a time stamp. Source ID depicts the sensor node that sends the packet while destination ID is the ID of next node that the packet is received. Normally, the destination ID is identified thanks to a routing table [31], a table that routes packets to a sink (or a gateway). These attributes compose a header of the packet. In addition to the header, the packet contains a data part which stores surrounding conditions such as light intensity, temperature, humidity, wind velocity, wind direction and hopper density.



HEADER						DATA					
ID	time Stamp	source ID	destID	location ID	data Size	density Hoppers	light Intensity	temperature	humidity	wind Velocity	wind Direction

Figure 10. Packet structurein the hopperobservationWSN.

A packet here depicts a structure to maintain a piece of spatial-temporal data. Indeed, time stamp illustrates the temporal aspect while location id describes the spatial one and the data part of the packet becomes data aspect. Figure 11 is an example of this piece of data. It can be translated as: at the time *01/02/2016 08:07:56 AM*, the sensor node *10* has *500 hoppers caught* at the temperature $29^o C$ and *5.5 km/h* wind velocity.



HEADER						DATA					
ID	01/02/2016 08:07:56 AM	...	...	10	...	500	...	29	...	5.5	...

Figure 11. Exampleof a piece of spatial-temporal data.

Consequently, a packet can be declared as the following C based pseudo code:

```
typedef struct{
  int ID;
  int sourceID, desID;
  // local ID of sensor node that senses data
  int location;
  //time stamp of sensing
  int timeStamp;
  int dataSize;
  float densityHoppers;
  float lightIntensity;
```

```
    float temperature, humidity;
    float velocityWind;
    float directionWind;
}LocalPacket;
```

Packet dissemination These behaviors compose of sensing surrounding conditions, packing into a packet and sending the packet to a gateway node (figure 12).

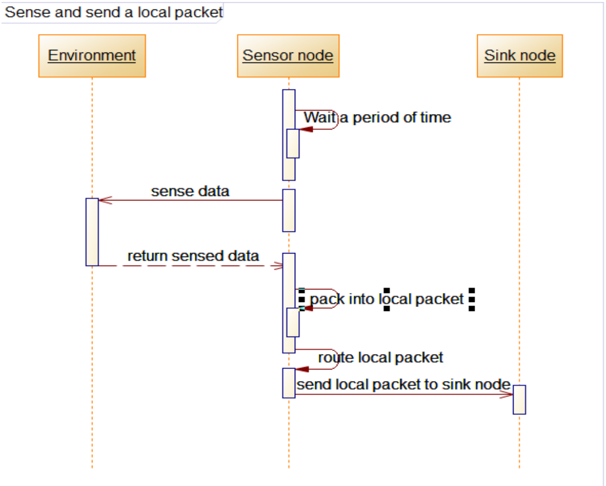


Figure 12. Procedure of sensing and sending a packet.

Figure 12 depicts behaviors of a sensor node in the hopper observation WSN. First, next execution time of a sensor node is identified by a timer. Next, the sensor node measures surrounding conditions and receives sensed data from environment, then these pieces of data are packed as a packet (as in figure 10). A pre-calculated routing table is used to route the local packet to its gateway where data is concentrated.

Timer In the WSN, timer is a mechanism calculating the time when nodes (sensor, sink and gateway) execute their tasks. This time can be figured out based on sensing times of sensor nodes and communication times to transmit packets to the gateway.

Assume that each sensor node contains n sensors, t_i is a duration of time to read sensor i. Therefore, $T_{read} = \{t_1, t_2, ..., t_m\}$ is a collections of times to read sensors in a sensor node.

Let $T_{sensing}(s)$ is the sensing time, a duration of time that a sensor node finish reading all its sensors. This time is calculated as:

$$\Rightarrow T_{sensing} = Max\{t_1, t_2, ..., t_m\} = Max\{T\} \quad (1)$$

To transmit data to the gateway, following factors are considered:

- f: the frequency used to transmit data (Hz).
- DataSize: sizes of a packet(bit).
- DataRate: the rate (speed) to transmit data ($Kbps$). This rate depends on the frequency used to transmit data.

The communication time $T_{local\_transmit}$ from one node to another node in the WSN is shown as below:

$$T_{transmit} = \frac{DataSize}{DataRate} \quad (2)$$

Therefore, the duration of time that a sensor finish executing its tasks (Figure 8) is calculated as:

$$T_{node} = T_{sensing} + T_{transmit} \quad (3)$$

Let $T_{to\_gateway}$ be a duration of time that the gateway finishes collecting data from all other nodes (figure 8), then $T_{to\_gateway}$ is estimated as:

$$T_{node} \leq T_{to\_gateway} \leq nT_{node} \quad (4)$$

where n is the number of sensor nodes of the WSN. Indeed, in the best case, all sensor nodes transmit their data directly to the gateway at the same time, then $T_{to\_gateway}$ is approximately T_{node}. However, in the worst case, sensor nodes perform sequentially, then $T_{to\_gateway}$ is around nT_{node}.

The interval time T between 2 next actions of a sensor node in the BPHSUN can be calculated as the maximum of $T_{to\_gateway}$ after adding an error-time Δt. Therefore, T is estimated as:

$$T = nT_{node} + \Delta t \quad (5)$$

$$\Rightarrow T = n(T_{sensing} + T_{transmit}) + \Delta t \quad (6)$$

Routing table Routing table [31] is a structure to store shortest paths from a node to other nodes in a WSN. In the BWSN, because automatic light trap sensor nodes located at fixed positions, routing table is pre-calculated and stored to disseminate data to the gateway.

Distance vector algorithm [31] is used to calculate routing tables for a WSN $G(V, E)$. The algorithm is shown as below:

ALGORITHM: Calculate a routing table for node v to other nodes in $G(V, E)$

INPUT: c[v, w]: the direct cost from v to w ($w \in V$).
OUTPUT: D[v, w]: distance between v and w ($w \in V$).
Next(v, w): next node to reach to w from v. ($w \in V$)

```
createRoutingtable(){
  for each (w in V)
        D[v, w] = 0;
  D[v, v] = _INFINITY;
  // Find route from v to others
  for each (w in V){
```

```
            if (w == v) continue;
            /* Select the shortest distance
               from v to its neighbors
                  after adding direct costs */
            minCost = D[v, w]; // old distance
            neighbor = −1;
            nextNode = Next(v, w);

            for each (j in v.getNeighbors()){
                  neighbor = j;
                  newCost = c[v,j] + D[j, w];
                  if (newCost < minCost){
                        minCost=newCost;
                        nextNode=neighbor;
                  }
            }

            // Update routing table
            D[v,w]=minCost;
            Next[v,w]=nextNode;
      }
}
```

The above algorithm is executed parallelly for all nodes $v \in V$, then paths from all node $v \in V$ to other nodes are found after 1 execution step. This procedure of calculating routing tables for all nodes is looped until distances are unchanged.

Example Let a WSN given by a graph $G_2(V_2, E_2)$ (figure 13). Assume that the direct cost from a node to its neighbors is 1.

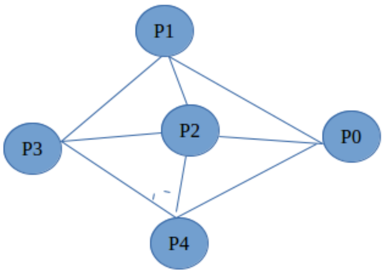
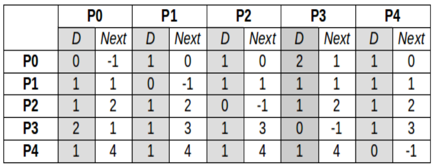


Figure 13. Exampleof a graph to calculate routingtables.

The following result are routing tables of all nodes in the graph G_2 after 2 execution steps. After 2 steps, routing tables of all nodes are unchanged, thus, they become final routing tables.

Routing packets It is a process of sending and receiving packets at each sensor node based on a routing table so that all packets are concentrated at the gateway.

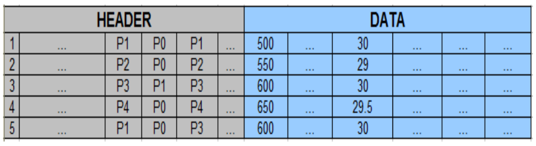
Before sensed data is transmitted, it is packed as a packet (as figure 10). The packet consists of an ID, source ID, destination ID, location, time stamp and other measured values of environmental factors. The attribute location depicts the local ID of the sensor node that senses these surrounding conditions.



	P0		P1		P2		P3		P4	
	D	*Next*	*D*	*Next*	*D*	*Next*	*D*	*Next*	*D*	*Next*
P0	0	-1	1	0	1	0	2	1	1	0
P1	1	1	0	-1	1	1	1	1	1	1
P2	1	2	1	2	0	-1	1	2	1	2
P3	2	1	1	3	1	3	0	-1	1	3
P4	1	4	1	4	1	4	1	4	0	-1

Figure 14. Routingtables after 2 steps.

Example: Assume that P1,P2,P3,P4 are sensor nodes and P0 is the sink node of the graph in figure **??**, according to routing tables in figure 14, collected data from P1, P2, P4 can send directly to P0 while P3 needs 2 hops to reach to P0 (P3 - P1 - P0). Figure 15 depicts a round how sensed data from sensor nodes reaches the sink node P0.



HEADER						DATA					
1	...	P1	P0	P1	...	500	...	30	...	...	...
2	...	P2	P0	P2	...	550	...	29	...	...	...
3	...	P3	P1	P3	...	600	...	30	...	...	...
4	...	P4	P0	P4	...	650	...	29.5	...	...	...
5	...	P1	P0	P3	...	600	...	30	...	...	...

Figure 15. Exampleof routinglocal packets.

To implement this mechanism, a sending buffer and a receiving buffer are maintained at each sensor node. These buffers are used in 2 following methods: sendLocalPackets() and receiveLocalPackets() to send and receive local packets at a node v. All nodes in a sub network at level 1 execute these methods concurrently.

Sending packets Sending packets takes place at sensor nodes of the WSN after they are granted execution times from the timer. First, environmental factors are sensed by sensors of a node and these pieces of sensed data are packed as a packet. Next, the packet is added to sending buffer of the node. The next step is to move all packets from the node's receiving buffer to its sending buffer in order to send to the gateway thanks to the routing table.

The algorithm is described as followed:

ALGORITHM: send local packets at a sensor node

INPUT: Node v, routing table t, receiving buffer *v.receiveBuff*

OUTPUT: sending buffer *v.sendBuf*

```
sendLocalPacket(){
 if (isExecutionTime()){
   if (!isLeader(v)){
     senseData();
//Sense data from environment
     createPacket(p); //Create a packet
```

```
    l = Leader(v);    //Find leader of v
    //Next hop of v is destination ID of p
    p.destinationID = t.Next[v][l];
    v.sendBuff.Add(p);
    for each (packet p in v.receiveBuff)
      v.sendBuff.Add(p);
    v.receiveBuff.Clear();
  }
 }
}
```

Receiving local packets When sensor nodes are granted execution times, they start receiving packets from their neighbors. Main idea of receiving packets at a node v is to locate in the sending buffer of each neighbor of v in order to find packets considering v as their destination IDs, then these packets are added to the receiving buffer of v.

ALGORITHM: receive local packets at a node

INPUT: Node v, routing table t
OUTPUT: receiving buffer *v.receiveBuf*

```
receiveLocalPacket(){
 if (isExecutionTime()){
   l = Leader(v); //Find leader of v
   for each (Neighbor j of v){
     for each (packet p in j.sendBuf){
       desID = p.destinationID;
       if (desID == v.ID){
         //Next hop of v is destination ID of p
         NextID =  t.Next[v][l];
         p.destinationID = NextID;
         v.receiveBuff.Add(p);
       }
     }
   }
 }
}
```

4.5. Cyber Physical System synchronization

Sensor nodes are deployed into some cells to sense meteorological conditions and to count the BPH density. In this problem, sensor nodes are automatic light traps which are able to sense these above factors and transmit these sensed values to a gateway. Circles in figure 5 depict communication ranges of sensor nodes while centers of these circles are sensor node localizations.

Normally, manual light traps are turned on at night to attract insects and insect densities are counted in the next morning. Therefore, automatic light trap sensor nodes almost sleep during day time. In addition, during night time, environmental conditions and hopper densities do not need to be sensed frequently; instead, it is suggested that they are calculated every interval time calculated in formula (6).

Synchronous points $t_0, t_1, ..., t_n$ (as in figure 4, section 3.6) between physical system and WSN system are calculated thanks to the interval time in formula (6). These synchronous points are necessary because the physical system is continuous while the WSN operates discretely. In addition, light traps are turned on at night to attract insects and insect densities are counted in the next morning; during night time, environmental conditions and hopper densities do not need to be sensed frequently because they do not change so much in a short period of time.

5. Bio-environmental surveillance network data model

A bio-environmental surveillance network plays important role in collecting data. Thanks to the current data collection and historical data, people can know better their solutions and can make suitable decisions. For example, in comparison with historical light trap data, the current trap data may indicate the peak season of BPH and people can propose a suitable cultivation time.

The data model of the bio-environmental surveillance network consists of 2 parts: meta-data model and data collection model. Meta-data model depicts the meta structure of a bio-environmental surveillance network, includes of the physical system and WSN system. On the other hand, data collection model is the model of collected data; it depends on what types of data people want to collect or sense.

5.1. Meta-data model

Figure 16 illustrates the meta-data model of a bio-environmental surveillance network. In this model, TCellularSystem, TCell, TSensorNode are important entities to depict 2 systems in a surveillance network: physical and network system. However, the main focus of the model is the entity TCell and its relationships.

Cell is a unit of the surveillance network. Each cell maintains a position in the image (xPos, yPos) and its geographical location (longitude, latitude, elevation). Each cell can be a Von Neumman type (4 neighbors), hexagon type (6 neighbors) or Moore type (8 neighbors).

The entity TCellSystem maintains the cellular system of the surveillance network. It can show the width and the height of a Von Neumman and Moore cell, or show the radius of a hexagonal cell. Besides, when a cellular system is created, a time stamp and a name is assigned to it. Each cellular system has an own ID.

Each pixel of the map of the physical system is stored in the entity TPixel. RGB color space is used to process image data. Besides, each pixel is in a fixed position (x,y)

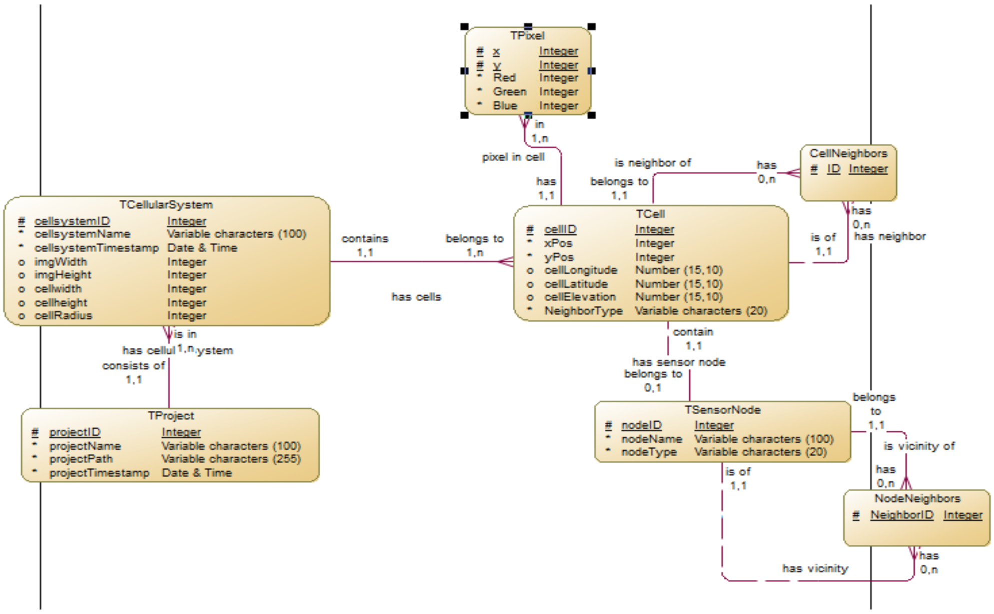


Figure 16. Data model for bio-environmental surveillance network.

in the image. A pixel belongs to a cell, but a cell can consist of many pixels.

The entity CellNeighbors maintains neighbors of cells. Each cell can have 4, 6, 8 neighbors. Each neighbor is assign an order from 0..3 (4 neighbors), 0..5 (6 neighbors), 0..7 (8 neighbors).

These above entities illustrate the physical system of a BIOSYN, however, entity TSensorNode depicts the WSN system in the BIOSYN. Each sensor node is assigned an ID, a name and type (sensor node, sink node, gateway). Each sensor node is in a cell while a cell can contain at most a sensor node.

In a more general case, people can have many cellular systems interacting each other, the entity TProject is introduced to solve it. A project can have many cellular system while a cellular system belongs to only one project.

5.2. Data collection model

Different surveillance applications have different data collection model. For instance, water level may be taken into account in the data collection model of a flooding surveillance. However, temperature, humidity, light intensity, wind, hopper density are considered as factors in a BPH surveillance data model.

To be simple, the data model consists of 2 entities TData, TSensingData which has relations with TCell and TSensor, respectively. Fields in these 2 entities are almost the same such as: time stamp and other environmental factors. The difference is the TData depicts the data generated from the physical system while TSensingData contains sampling sensing data from surrounding conditions.

For example, the data collection model of a BPH surveillance network is as followed (figure 17):

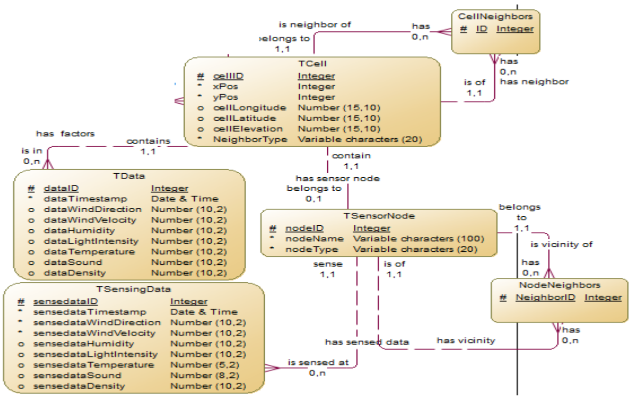


Figure 17. Data model for BPH surveillance.

6. Implementation

6.1. Workflo

The implementation of a BIOSYN contains 3 important parts: data structure, states and behaviors. Firstly, data structure (cells in figure 2) is generated from geographic data. Next, states and behaviors are implemented in CUDA [20] to illustrate synchronous characteristics in the model.

CUDA is chosen for implementing BPHSYN since the parallel programming paradigm of CUDA is well-suited for the model's concurrency.

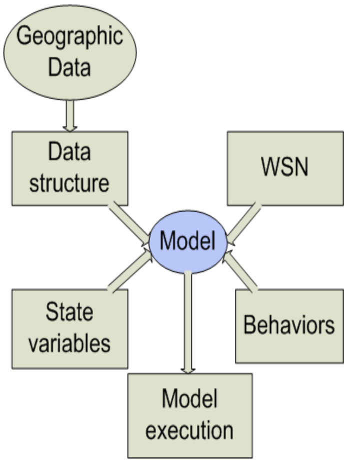


Figure 18. Workflow for modeling physical and WSN system.

6.2. Physical system data structure

Graph. The following CUDA code is the definition of the physical system graph. Supposes that the physical system uses hexagonal cellular automata.

```
typedef struct{
  int x, y;
}Point2D;
typedef struct{
  int xPos, yPos; //offset coordinates
  Point2D Corners[6]; // 6 corners
  int Neighbors[6]; // 6 neighbors
}Hexagon;
// Honeycomb structure
typedef Hexagrid Hexagon[N];
```

State variables. Each node in the graph represents a process in a synchronous network. This process maintains a structure to store its state variables by the time passing. The following code is the definition of a state variable in the case study BPH surveillance network:

```
typedef struct{
  int xPos, yPos; //position
  float          riceAges;    // rice ages
  InsectDensity densityBPH;
  float          windVelocity;
  Direction      windDirection;
}State;
```

6.3. Wireless Sensor Network data structure

Sensor nodes are distributed in some cells of the graph. The following code describes the definition of the WSN in a bio-environmental surveillance network:

```
typedef struct{
  int node;
  float distance;
}Neighbor;
typedef struct{
  //offset coordinates of the hexa cell
  int xPos, yPos;
  // neighbors of sensor node
  Neighbor Neighbors[MAXFLOW];
}Channel;
typedef Network Channel[N]; // WSN
```

6.4. Behaviors implementation

Behaviors of the physical system and WSN system depends on different surveillance networks. For example, these behaviors of a BPH surveillance network are implemented as descriptions in 4.3 and 4.4.

7. Experiment

7.1. Data used

The simulation of the BPH synchronous network uses data collections in Cantho city (figure 5), a typical rice city in the Mekong Delta. Current light traps (8 traps till 2015) are considered as sensor nodes, automatic light trap sensor nodes, in the simulation (circles in figure 5). Hexagonal cellular automata is used by dividing the area of Cantho into hexagonal cells with approximately $0.18km^2$ each.

7.2. Tools used

Some tools are used in this experiment. A self developed tool HexGen is used to generate codes of hexagonal cellular automata synchronous networks in Cuda. Besides, the map of Cantho city is processed by the tool PickCell in the framework NetGen [18]. Behaviors of the BPHSYN are implemented in CUDA to run the simulation on the NVIDIA card GeForce GTX 680 1.15GHz with 1536 CUDA Cores (8 Multiprocessors x 192 CUDA Cores/MP).

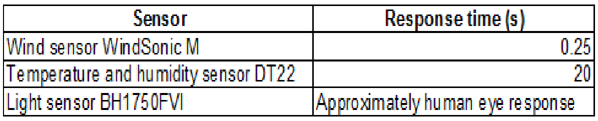
7.3. Sensor node interval time calculation

According to Formula 6 the sensor node interval time of the BPHSUN depends on sensing times and communication times.

Actually, sensing time of a sensor is the response time of that sensor. This time relies on type of sensors as well as concrete surrounding conditions. A sensor node here is an automatic light trap [35] consists of following sensors shown in figure 19. This figure also depicts the response time of each sensor [32][33][34].

Therefore, the sensing time is $T_{sensing}$ = 20s. There are 8 sensor nodes ⇒ n = 8.

LORA technology [28] is used to transmit data in the BPH surveillance network in Mekong Delta[35]. The board Semtech SX1276 [28] is used since it is suitable for allowance frequencies in Vietnam. The specification of this board shows that it has 0.018-37.5 kbps data



Sensor	Response time (s)
Wind sensor WindSonic M	0.25
Temperature and humidity sensor DT22	20
Light sensor BH1750FVI	Approximately human eye response

Figure 19. Response times of sensors in a light trap sensor node

rate, the frequency 433Mhz. Assume that the board is configured to work with 11kbps data rate.

According to 10, the size of each packet is 48 bytes ($DataSize = 48$).

$$T_{transmit} = \frac{DataSize}{DataRate} = \frac{48 * 8}{11 * 1000} = 0.384s$$

According to formula 6, the interval time T is calculated as:

$$\Rightarrow T = 163.072 + \Delta t(s)$$

Because environmental factors do not change so much during a short period of time, $\Delta t = 1636.928$s is chosen $\Rightarrow T$=1800s = 30 minutes. Thus, the interval time between 2 adjacent actions of a sensor node is 30 minutes. If a light trap works 4 hours every night, the sensor node senses environment and transmits data 9 times.

7.4. Scenario: physical system and network system synchronization

This simulation depicts the interaction between the insect physical system and WSN system in the BPH surveillance network. Sampling data from sensor nodes play as inputs for the insect physical system. On the other hands, when the insect physical system occurs, the collected data from sensor nodes illustrates the damage level caused by BPHs.

In the beginning, the WSN gives environmental conditions at 8 locations of sensor nodes. The temperature is 30^oC (the common temperature in Mekong Delta). The wind direction is 2 (the direction from the **center cell** to **cell 5** in figure 8 with the velocity is 5km/h (approximately 10 hexagonal cells/h). That means BPHs can transmit to the cell distance 10 from the source cell in one hour. Besides, almost rice fields in Cantho are almost young and suffer lightly from hoppers (light infection color of rice fields and sensor nodes as in figure 20).

The density of hoppers tends to increase in the next days. Due to wind, hoppers spread in other rice fields of Cantho and their densities it rises gradually until the 5^{th} day. On this day, hopper burns appear in almost communes in Cantho while few places are light or medium infection (figure 22). The infected area reaches maximum value on the 13^{th} day, then it begins decreasing. On the 30^{th} day, it almost becomes normal on the rice fields and in sensor nodes.

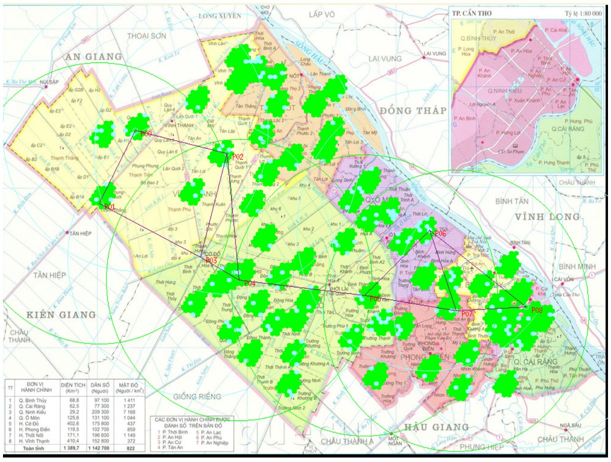


Figure 20. BPHs in rice field and sensor nodes in the beginning.

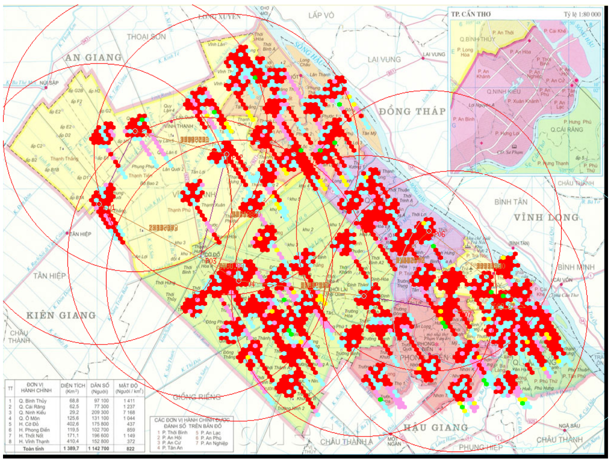


Figure 21. Hopper burns on the 13^{th} day.

Figure 22 illustrates tendencies of BPHs in rice fields and automatic light trap sensor nodes during this simulation. Starting with light infections in the beginning, densities of hoppers increase gradually and reach the peak point at the $13^{th}, 14^{th}, 15^{th}$ day. From that day, they decrease slowly and they almost become normal densities at the 30^{th} day.

Figure 22 also describes that the density of hoppers caught in automatic light trap sensor nodes is ratio with that of hoppers in rice fields. In other words, data collection in sensor nodes can be used as an indication for the hopper infection in rice field. However, there is not many investigation about the relation between them.

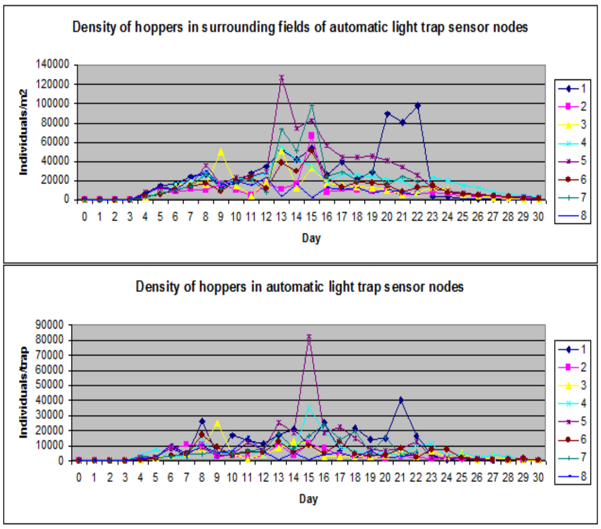


Figure 22. Rice field and automatic light trap sensor node data collection after 30 days of simulation.

8. Conclusion

We have described the a bio-environmental surveillance network BIOSYN as synchronous networks. In our work, a BIOSYN is an interaction of a physical process, modeled as a synchronous network, and a synchronous WSN. Environmental factors sensed in the WSN provide inputs for the execution of the bio-environmental physical system. On the other hands, the effect of the bio-environmental phenomena is sampling measured by sensor nodes. These composes a feed back loop between the physical system and the network system.

Cellular automata are considered of the bio-environmental surveillance network topology, especially in the physical system. Indeed, bio-environmental phenomena behave in an environment which is divided as cellular cells of which each has 4, 6, 8 neighbors. These cells compose a synchronous network which is modeled as a graph by considering each cell is as a node and two vicinity cells compose an edge. In addition, concurrent characteristics of the network are depicted as node behaviors.

The mesh network is used to disseminate data via the WSN network. The WSN is considered as a graph of which nodes are sensor nodes and edges are composed thanks to sensor node communication ranges. A timer is used to schedule sensor node behaviors in synchronous rounds to disseminate data to a gateway. A database schema is also introduced to maintain historical sensor node sampling data for post processing.

9. Acknowledgements

The work of this paper was in the scope of the "OBSNET - Optimizing the Brown Planthopper surveillance network using automatic light traps in the Mekong Delta region" project of the Ministry of Education and Training, Vietnam. The first author was funded by the program 911 of Ministry of Education and Training, Vietnam.

References

[1] Bao Hoai Lam, Hoang Van Tran, Hiep Xuan Huynh and Bernard Pottier (2015) *Synchronous network for insect surveillance* (ACM), in Proceedings of the Sixth International Symposium on Information and Communication Technology, Hue City, Vietnam, December 3-4, 2015, ISBN 978-1-4503-3843-1, pp. 27–34.

[2] Kelvin F. Conrad, Richard Fox and Ian P. Woiwod (2007) *Monitoring biodiversity: measuring long-term changes in insect abundance* (Cabi Publisher), Insect conservation biology, ISBN 9781845932541, pp. 203–225.

[3] Woiwod, I. P. and Harrington, R. (1994) *Flying in the face of change: The Rothamsted Insect Survey* (Wallingford: CAB International), in Long-term Experiments in Agricultural and Ecological Sciences (ed. R. A. Leigh & A. E. Johnston), pp. 321–342.

[4] Hoai Bao Lam, Tai Tan Phan, Long Huynh Vuong, Hiep Xuan Huynh, and Bernard Pottier (2013) *Designing a brown planthoppers surveillance network based on wireless sensor network approach*, in: ISCRAM (Information Systems for Crisis Response and Management) Vietnam 2013 Conference.

[5] Bao Hoai Lam, Hiep Xuan Huynh, Mahamadou Traoré, Pierre Yves Lucas, and Bernard Pottier (2014) *Monitoring environmental factors in Mekong Delta of Vietnam using Wireless Sensor Network approach*, in: 8th International conference on Simulation and Modelling in the Food and Bio-Industry FoodSim'2014, ISBN 978-90-77381-84-7, pp. 71–78.

[6] Cuong H. Phan, Hiep Xuan Huynh, and Alexis Drogoul (2010) *An Agent-Based Approach to the Simulation of Brown Plant Hopper (BPH) Invasions in the Mekong Delta* (IEEE), in: 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), Hanoi, Vietnam, 1–6.

[7] Nguyen, Vinh Gia Nhi and Huynh, Hiep Xuan, Vo, Tung Thanh, and Drogoul, Alexis (2011) *On Weather Affecting to Brown Plant Hopper Invasion Using an Agent-based Model* (ACM, New York, NY, USA), in: Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES '11, ISBN 978-1-4503-1047-5, pp. 150–157.

[8] Viet Xuan Truong, Hiep Xuan Huynh, Minh Ngoc Le, and Alexis Drogoul (2012) *Modeling a Surveillance Network Based on Unit Disk Graph Technique - Application for Monitoring the Invasion of Insects in Mekong Delta Region* (Springer), in: PRIMA 2012: Principles and Practice of Multi-Agent Systems - 15th International Conference, Kuching, Sarawak, Malaysia, pp. 228–242.

[9] http://www.healthyforestpartnership.ca/

[10] Marc Rhainds, and Stephen B. Heard (2014) *Sampling procedures and adult sex ratios in spruce budworm*, Entomologia Experimentalis et Applicata.

[11] Marc Rhainds (2015) *Wing wear and body size measurements of adult spruce budworms captured at light traps: inference on seasonal patterns related to reproduction* (Springer Japan), in Applied Entomology and Zoolog, Volumn 50, no. 4, pp. 477–485, ISBN 0003-6862.

[12] K. L. Heong, and B. Hardy (2009) *Planthoppers: new threats to the sustainability of intensive rice production systems in Asia* (International Rice Research Institute), ISBN 978-90-77381-84-7.

[13] Edward A. Lee (2010) *CPS Foundations* (ACM), in Proc. of the 47th Design Automation Conference (DAC), pp. 737–742.

[14] Lynch, Nancy A (1996) *Distributed Algorithms* (Morgan Kaufmann Publishers Inc), ISBN 155860348.

[15] WH. Reissig, E.A Heinrichs, J.A. Litsinger, K. Moody, L. Fiedler, W.Mew, and A.T. Barrion (1986) *Illustrated guide to integrated pest management in rice in tropical Asia*, IRRI, Philippines.

[16] S. Cheng, J. Chen, H. Si, L. Yan, T. Chu, C. Wu, J. Chien, and C. Yan (1979) *Studies on the migrations of brown planthopp ers nilaparvata lugens*, Std. Acta Entomol. Sinica, 22: pp. 1–21 (Chinese, English summary).

[17] Yingshu Li, My T. Thai, and Weili Wu (2008) *Wireless Sensor Networks and Applications* (Springer).

[18] Bernard Pottier, and Pierre-Yves Lucas (2015) *Dynamic networks "NetGen: objectives, installation, use, and programming"*, Université de Bretagne Occidentale, https://github.com/NetGenProject.

[19] Guillaume Kremer, Jimmy Osmont and Bernard Pottier (2009) *A Process Oriented Development Flow for Wireless Sensor Networks*, International Workshop on Smalltalk Technologies - ESUG'09 Brest.

[20] NVDIA, https://developer.nvidia.com/cuda-zone.

[21] GPU, http://www.nvidia.com/object/what-is-gpu-computing.html.

[22] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Federate Interface Specification, IEEE Std 1516.1-2010, pp. 1–378.

[23] Dufay, C (1964) *Contribution a l'Étude du phototropisme des Lépidopteres noctuides*, Annales des Sciences Naturelles Zoologie, Paris, vol 12. no 6, pp. 81–406.

[24] Stephen Wolfram (1994) *Cellular Automata and Complexity* (Westwiew Press, Perseus Books Group), ISBN: 0-201-62716-7.

[25] Stephen Wolfram *Statistical mechanics of cellular automata* (American Physical Society), Reviews of Modern Physics vol. 55, pp. 601–644.

[26] Ho Van Chien, Nguyen Huu Huan, and Le Quoc Cuong (2012) *Escape Strategy can successfully manage BPH and virus disease in the Mekong*, http://ricehoppers.net/2012/09/escape-strategy-can-successfully-manage-bph-and-virus-disease-in-the-mekong

[27] José Cecílio, and Furtado Pedro (2014) *Wireless Sensor Networks: Concepts and Components* (Springer International Publishing), in Wireless Sensors in Heterogeneous Networked System, Computer Communications and Networks, pp. 5—25.

[28] LORA technology, http://www.semtech.com/wireless-rf/rf-transceivers/sx1276/.

[29] G. Lasnier, J. Cardoso, P. Siron, C. Pagetti, and P. Derler (2013) *Distributed simulation of heterogeneous and real-time systems*, in Distributed Simulation and Real Time Applications (DS-RT), 2013 IEEE/ACM 17th International Symposium on, pp. 55—62.

[30] Truong Chi Quang, Vo Quang Minh, Nguyen Thi Hong Chau, and Ho Van Chien (2013) *Managing the Brown Planthopper caught by light traps for supporting of rice cultivation in the Mekong Delta*, in National GIS application conference (in Vietnamese).

[31] Vutukury, S.; Garcia-Luna-Aceves, J.J. (2001) *MDVA: a distance-vector multipath routing protocol*, in INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.1, no., pp.557–564 vol.1, doi: 10.1109/INFCOM.2001.916780.

[32] WindSonic M datasheet, http://gillinstruments.com/data/datasheets/windsonic-m.pdf?v=01.2014

[33] DT22, http://www.dfrobot.com

[34] BH1750FVI, http://www.electronic-circuits-diagrams.com

[35] Ky Minh Nguyen, Bao Hoai Lam, Tuyen Phong Truong, Hieu Minh Thi Pham, Chien Van Ho, Bernard Pottier and Hiep Xuan Huynh (2015) *Automatic hopper light trap*, in Cantho University Journal of Science, pp. 168–178, Cantho University.

[36] Hefeeda M., Bagheri M. (2007) *Wireless Sensor Networks for Early Detection of Forest Fire*, in Mobile Adhoc and Sensor Systems, MASS 2007, pp. 1-6.

[37] Government of Canada http://cwfis.cfs.nrcan.gc.ca/