Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

SIMULATION OF LARGE WIRELESS SENSOR NETWORKS USING CELL-DEVS

Blerim Qela

School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, Canada

Gabriel Wainer

Department Of Systems and Computer Engineering.
Centre of Visualization and Simulation (V-Sim)
Carleton University
Ottawa, ON, Canada

Hussein Mouftah

School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, Canada

ABSTRACT

The advancement of electronic sensing devices, microcomputers and wireless communication devices has lead to creation of new smart sensors, which can monitor actuate, compute and communicate. Typically, these sensors are deployed in non-deterministic mode (randomly) when deployed in large numbers. These sensor devices have the capability to self-organize into the so-called Wireless Sensor Networks (WSN). WSN are ad-hoc networks, consisting of these spatially distributed sensing and processing devices. We introduce a model and a simulation study of these Large Wireless Sensor Networks (WSN) by implementing the Topology Control Algorithm. We use the Cell-DEVS formalism, which enables efficient execution of cellular models. Thereafter, we observe and evaluate the behavior of sensor nodes and entire WSN from the simulation results obtained, under different test scenarios.

1 INTRODUCTION

The emergence of powerful embedded micro-computer systems for wireless sensor networks provides a good ground for creation of new smart sensor systems, which can be useful to further promote new scientific endeavors and enhance our lives.

The wireless sensor devices can monitor actuate, compute and communicate, yet are small in size and low in cost. The WSN are ad-hoc networks, consisting of these spatially distributed sensing and processing devices (Cunha, Silva, and Loreiro, 2005; Healy, Newe, and Lewis 2008). WSN are used in many different applications, such as medicine, transportation and urban monitoring, traffic control, military, environment and habitat monitoring, energy management, smart homes, industrial applications, etc. The effectiveness of WSN is not just in their monitoring, actuating, computing and communications capabilities: with the added processing power, analog and digital ports, transceivers and memory, they can self-organize and communicate in the deployed area (as depicted in Fig.1). Their processing power is limited however, and WSNs are usually deployed in large numbers and their load is shared accordingly.

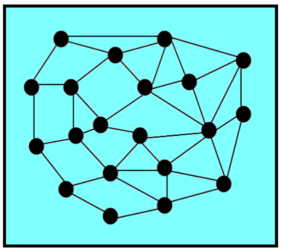


Fig.1 - Sensor nodes self-organized

Due to the fact that sensor nodes have limited bandwidth, computing power and limited energy resources, one of the constraints in WSN is the energy efficiency of sensor nodes (i.e., their power consumption). There are many different approaches to solving the energy efficiency problem, such as the Topology Control Algorithm (Cunha, Silva, and Loreiro 2005;

Shen, and Guo 2007; Ye et al. 2002). The objective of this algorithm is related to the efficiency of WSN network, and it focused on how to increase its lifetime. The rationale of the network topology control algorithm is to reduce the number of redundant sensor nodes monitoring a particular region, hence increasing the efficiency and lifetime of the WSN network. In a nutshell, topology control exploits the redundant deployment of the sensor nodes, overcoming their energy limitations by restricting the set of nodes which are considered neighbors of a given node, while making sure that sensing area is still covered by a sufficient number of sensors. Furthermore, it reduces interference problems (which are noticeable when large number of sensor nodes are active).

We here introduce a model and a simulation of Wireless Sensor Networks (WSN) by implementing the Topology Control Algorithm using the Cell-DEVS formalism (which enabled efficient execution of this cellular model). The complexity of the problem can be simplified using the CD++ toolkit and the Cell-DEVS formalism (implementing this simulation in a other high level programming language is much more complex).

The rest of the paper is organized as follows: we first give an overview of the Cell-DEVS formalism, followed by the WSN model definition. Thereafter, we observe and evaluate the behavior of sensor nodes and entire WSN from the simulation results obtained, under different test scenarios. We then describe and discuss the simulation results, analysis and discussion of the initial and improved results for different scenarios.

2 MODEL DEFINITION

Due to the complexity of the model under study, we used Cell-DEVS (Wainer 2009) and the CD++ toolkit (Wainer 2002), as an efficient way to model and simulate cellular models (Chopard and Droz1998), in our case the WSN topology problem. Cell-DEVS is an extension to the DEVS formalism (Zeigler, Kim, and Praehofer 2000), which has been used to model systems that can be represented as cell spaces. A Cell-DEVS model is represented as a cell space, where each cell is represented as an atomic DEVS model. Each cell is connected to the local neighboring cells. A delay mechanism in each cell (transport delay or inertial delay) is used to delay the propagation of state change events through the cell space, providing the means for defining complex temporal behavior. An Atomic Cell-DEVS can be defined as follows:

$$TDC = < X, Y, I, S, \theta, N, d, \delta int, \delta ext, \tau, \lambda, D >$$

Where X is the set of external input events; Y is the set of external output events; I represents the definition of the model's modular interface; S is the set of possible states for a given cell; θ is the definition of the cell's state variables, N is the set of values for the input events; d is the delay of the cell; δint is the internal transition function; δext is the external transition function; τ is the local computing function; λ is the output function, and D is the duration function.

A Coupled Cell-DEVS model is built by connecting a number of Atomic Cell-DEVS models together into a cell space of any shape (including 2D and 3D cell spaces). The borders of the cell space can be either wrapped, in which case the cells at the border from one side of the cell space are considered neighbors to the cells at the border on the opposite side of the cell-space, or non-wrapped, in which case the border cells must have special rules defined by the modeler. A formal definition of Coupled Cell-DEVS is:

$$GCC = < Xlist, Ylist, I, X, Y, n, \{t1,\ldots,tn\}, N, C, B, Z, select >$$

Where Xlist is the input coupling list; Ylist is the output coupling list; I represents the definition of the model's modular interface; X is the set of external input events; Y is the set of external output events; n is the dimension of the cell space; {t1,….,tn}is the number of cells in each of the dimensions; N is the neighborhood set; C is the cell space; B is the set of border cells; Z is the translation function; and select is the tie-breaking function;

CD++ is an M&S toolkit that implements DEVS and Cell-DEVS theory ((Wainer 2009; Wainer 2002; Zeigler, Kim, and Praehofer 2000). Atomic models can be defined using a state-based approach (coded in C++ or an interpreted graphical notation), while coupled and Cell-DEVS models are defined using a built-in specification language. CD++ also includes an interpreter for Cell-DEVS models. The model specification includes the definition of the size and dimension of the cell space, the shape of the neighborhood and borders. The cell's local computing function is defined using a set of rules with the form: POSTCONDITION DELAY {PRECONDITION}. This indicates that when the *PRECONDITION* is satisfied, the state of the cell will change to the designated *POSTCONDITION*, whose computed value will be transmitted to other components after consuming the *DELAY*.

We have used these facilities to create an advanced model of WSN, in which we can analyze the WSN topology problem. This problem is closely related to the minimum configuration of nodes for fully operational WSN, taking into account sensor node energy limitations for a long lasting - survivable WSN networks. In our model, we have structured the area as a two-dimensional cell space of size ***n* x *n***, where every cell represents one sensor node. The WSN considered is such as all the sensor nodes have same properties (homogeneous WSN) and flat (i.e. no hierarchy among nodes). Each node can have up to 8 neighbors and there are ***3 possible states*** for each cell (*active*, *stand-by*, and *inactive*). For this model, Moore's neighborhood is adopted (i.e., the origin cell and its 8 close neighbors). As a result, the Cell-DEVS simulation model, gives an insight into an elegant way of implementation the Topology Control Algorithm for a large WSN; in our particular case, addressing the issue of sensor node energy conservation to achieve a longer lifetime operation of the WSN.

During the *active mode* of operation, the node emulates an active sensor within the WSN (i.e. performs processing tasks, hence using energy which decreases with time). In the beginning of simulation, all the sensor nodes deployed, have the same amount of energy which decreases as time progresses while node is in active mode (maximum energy consumption) or in stand-by mode (minimum energy consumption). During the *stand-by mode*, sensor node is consuming a minimal amount of energy; it wakes up randomly in order to see if other close by nodes are already sensing/monitoring the predefined neighborhood area. If less than two sensor nodes within the neighborhood are *active*, the cell goes again into *stand-by* mode; otherwise it becomes *active*. After the entire energy of a cell is consumed, the cell becomes *inactive*; this process continues until all the sensor nodes' energy is consumed (i.e. all the cells are inactive).

The ***n* x *n*** cells in the cell space considered were organized in two planes reflecting a three dimensional space implemented to meet the basic constraints (of the defined WSN Topology problem) while exploring and capturing the main tasks of the problem considered. Each sensor node starts to operate with a fixed amount of energy. In this model the energy levels adopted based on (Cunha, Silva, and Loreiro 2005; Ye et al. 2002) are the following:

- WSN sensor energy (at the beginning of operation) = 0.8 J
- The energy of an WSN sensor node in ***active mode*** decreases by 0.0165 J every time step (in our case every 1 sec)
- The energy of a WSN sensor node in ***stand-by mode*** decreases by 0.00006 J every time step (in our case every 1 sec)

The WSN sensor node possible states are:

Plane 0:

2 - WSN sensor node is in ***active mode*** within the neighborhood (WSN sensor coverage area)
1 - WSN sensor node is in ***stand-by mode*** within the neighborhood (WSN sensor coverage area)
0 – WSN sensor node is ***passive*** (i.e. energy of a node is consumed)

Plane 1:

0.8 - initial energy level of WSN sensor nodes
-1 - WSN sensor node is ***passive*** (i.e. energy of a node is consumed)
Neighborhood ={ (-1,-1,0) (-1,0,0), (-1,1,0), (0,-1,0), (0,0,0), (0,1,0)(1,-1,0),(1,0,0),(1,1,0), (0,0,1), (0,0,-1) }

Plane 0 contains the different deployed sensor nodes, and it is used to be observed throughout the simulation of the model. Plane 1 was used as "memory" for keeping track of the energy levels of active and stand-by nodes. The correct decrease of energy level and node behavior in the Plane 0 is interrelated to Plane 1 which contains the sensors energy information throughout the simulation.

The Plane 0 areas are organized as follows:

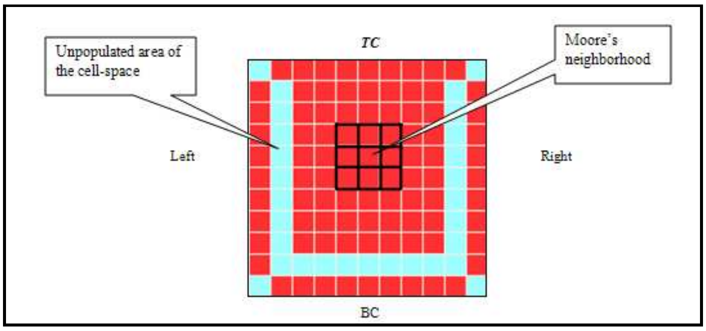


Fig.2 - Plane 0 organization (zones)

TL	Top left rule, sensor node in the top left corner, i.e. origin coordinate (0, 0)
TC	Top center rule, sensor node coordinates (0, 1) to (0, 9)
BC	Bottom center rule, sensor node coordinates (10, 1) to (10, 9)
TR	Top right rule, sensor node coordinate (0, 10)
BL	Bottom left rule, sensor node coordinate (10, 0)
BR	Bottom right rule, sensor node coordinate (10, 10)
Right rule	sensor node coordinates (1, 10) to (9, 10)
Left rule	sensor node coordinates (1, 0) to (9, 0)

The rest of a cell space is WSN rule, i.e. local transition rule of the model.

The initial model was organized as shown in Fig.2. The partitioning of the cell space into 'zones' was done to experiment with the problem at hand (initially) by isolating the 'zones'. Hence, easing the task to be solved, by observing sensors' operations at different 'zones' (and their interaction) prior to optimizing the final solution.

When compared to the initial model, the improved model (refer to Fig.3) are the following:

- Entire cell-space can be populated with sensor nodes
- There are no limitations to functionality of the model (Plane 0 organization)
- Cell space is not divided into zones
- Enables more flexibility to the initial model (reduces significantly the code size)
- Randomness was implemented within the model (it enables us to represent more closely the real-world applications related to WSN)

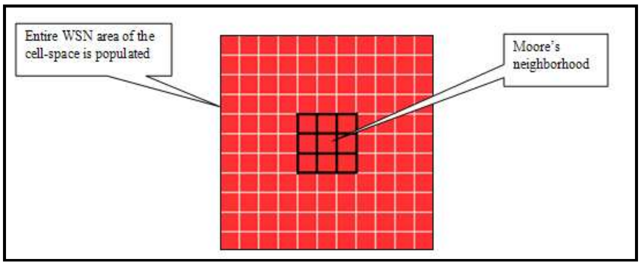


Fig.3 - Plane 0 organization (improved model)

In addition to the existing states present in the initial WSN model (described earlier), an additional state was implemented in Plane 0 of the improved WSN model:

3 – WSN sensor node is typically in ***stand-by mode***; however, nodes randomly become active if one or two nodes are active Within the neighborhood (Moore's neighborhood coverage area); node goes back to ***stand-by mode*** if the condition is true (i.e. one or two neighborhood sensors are active), otherwise it remains in the ***active mode***.

2.1 WSN behavior definition

In Plane 0, each Moore's neighborhood (consisting of 9 cells) typically is covered by one active cell (with the value of 2) and the rest of the stand-by cells have a value of 1, whilst the passive cells have a value of 0. At the initial stage all the WSN sensor nodes deployed within the WSN network area (represented by cell space) are active for several time steps until they configure themselves to active and stand-by mode nodes.

In Plane 1, for every time step, the energy of active cells is decreased; the amount of energy available $\boldsymbol{x}$, for any active and stand-by mode cells before cell dies is $0 \leq \boldsymbol{x} \leq 0.8$. Each cells energy level in Plane 1 serves as a reference (memory) of the Plane 0 active and stand-by mode cells (represented by value of 2 and 1 respectively). Plane 0 refers to Plane 1 by (0, 0, 1), while Plane 1 refers to Plane 0 by (0, 0,-1) neighborhood coordinate (refer to Fig.4). The neighboring cells which are in a stand-by mode are represented by the value of 1 within the cell, and are also decreased in negligible quantities during the stand-by mode; The active cells become passive after their energy is consumed and are replaced by the neighboring cells which are yet alive (cells currently in stand-by mode).

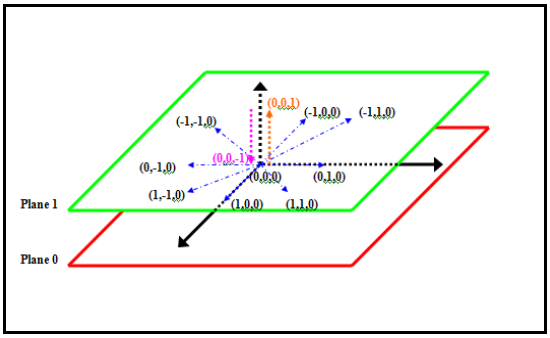


Fig.4 - Cell space definition

The neighboring cells which are in a stand-by mode are represented by the value of 1 within the cell, and are also decreased in negligible quantities during the stand-by mode; The active cells become passive after their energy is consumed and are replaced by the neighboring cells which are yet alive (cells currently in stand-by mode). The cell-space in Plane 0 defines sensor nodes, whilst in Plane 1 energy level corresponding to each node (refer to Fig.4).

The algorithm steps through each neighborhood and decides on which nodes stays active and which is to be configured as a stand-by nodes; where each neighborhood of cells is checked for present active cells by comparing the present cell's value with the residual energy (which is check referenced in Plane 1) until they becomes passive (represented by value of 0 in the cell space); Afterwards, one of the stand-by neighboring cell is assigned to becomes active, and this process continues until all the cells become passive (i.e. the energy resources of all sensors is consumed). Plane 1 updates the energy levels of each cell during the simulation as per the specifications.

The WSN simulation model provides closer approximation to the WSN topology algorithm by implementation of randomness within the WSN model; where randomness, redundancy and configuration of nodes play a significant role in reflecting the key factors in deployed networks, such as a network lifetime, coverage area and ratios of active and stand-by sensors at specific points in time. Randomness was implemented by adding another rule (refer to the rule 3 below) to the model, where stand-by cells randomly become active if only one or two active neighbors are active. The actual results obtained with the improved model, more closely reflect the real-world scenarios and provide better insight into the WSN topology problem.

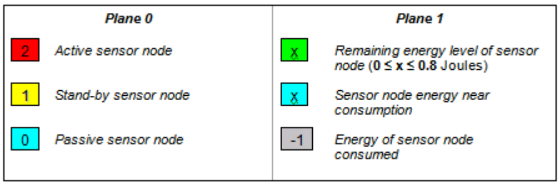
The problem could be reduced and coded with less than 30 lines of code utilizing the CD++ toolkit and Cell-DEVS, as shown in the following figure

```
[WSN]
type : cell                   dim: (33, 33, 2)                 delay : transport
border : nowrapped
neighbors :  (-1,-1,0)  (-1,0,0)  (-1,1,0) (0,-1,0)   (0,0,0)   (0,1,0)
neighbors :  (1,-1,0)   (1,0,0)   (1,1,0)  (0,0,1)    (0,0,-1)

localtransition : WSN-rule
[WSN-rule]
rule : { (0,0,0) - 0.0165 }  1000 {cellpos(2) = 1 and  (0,0,-1) = 2 }
rule : { (0,0,0) - 0.00006 } 1000 {cellpos(2) = 1 and (0,0,-1) =1  }
rule : { (0,0,0) - 0.0165 }  1000 {cellpos(2) = 1 and  (0,0,-1) = 3 }
rule : 3 1000 {cellpos(2)=0  and (0,0,1)>0 and
               ( (0,0,0)=1   and (statecount(2)=2 or statecount(2)=1) and randInt(30)=11 ) }
rule : 2 1000 {cellpos(2)=0  and (0,0,0) !=0 and (0,0,0) !=3 and (0,0,1) > 0 and (0,-1,0)!=2
             and (1,-1,0)!=2 and (1,0,0)!=2 and (1,1,0)!=2 }
rule : 1 1000 {cellpos(2)=0  and (0,0,0) !=0 and (0,0,1)>0 and
               ( (0,-1,0)=2  or (1,-1,0)=2 or (1,0,0)=2 or (0,1,0)=2 or (1,1,0)=2 or
              (cellpos(2)=0  and (0,0,0)=3) or (-1,-1,0)=2 or (-1,0,0)=2 or (-1,1,0)=2 ) }
rule : 0 1000 {cellpos(2)=0  and ((0,0,0)=1 or (0,0,0)=2 or (0,0,0)=0)  and (0,0,1) <= 0 }
```

3 SIMULATION RESULTS

We executed numerous tests, and in this section we present some of the simulation results obtained and discuss their meaning. The Fig.5 shows the graphical representation of the sensor node states defined in the simulation model:

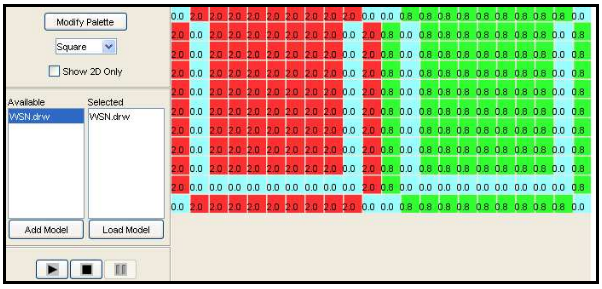


Plane 0		***Plane 1***	
2	*Active sensor node*	x	*Remaining energy level of sensor node* (**0 ≤ x ≤ 0.8** Joules)
1	*Stand-by sensor node*	x	*Sensor node energy near consumption*
0	*Passive sensor node*	-1	*Energy of sensor node consumed*

Fig. 5 - Possible sensor states of the initial WSN model

On the left side of the Fig.5 we have the Plane 0 configuration, where **2** - indicates that a sensor node is ***active***, **1** – indicates that a sensor node is in ***stand-by mode*** and **0** - indicates that a sensor node is in ***passive state*** (i.e. energy of a node is consumed); while on the right side of the Fig.5 we have the Plane 1 configuration, where **x** – is the energy level of WSN sensor nodes (green cells are the sensor nodes with enough energy, while blue color signifies that the sensor nodes are close to dying), **1** - WSN sensor node is ***passive*** (i.e. energy of a node is consumed)

The first simulation results presented are shown in Fig.6 below. Based on the specification, we can see the sensor nodes active at the beginning of the simulation (red cells), while the light blue cells represent the unpopulated zones within the cell space.



0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	2.0	0.8	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0	0.8
2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8
0.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	0.0	0.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.0

Fig.6- Plane 0 (left, red) and Plane 1 (right, green) prior to execution

The following Fig.7 shows the WSN network during the process of reconfiguration, where nodes are trying to form a structure and every neighbor is trying to set some sensors in active mode while others remain in stand-by. The Plane 1 stores the energy level of each sensor node, which can be monitored.

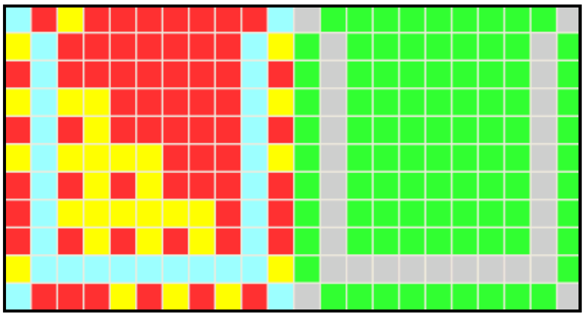


Fig.7 - Snapshot of simulations results after 6 time steps

Fig.8 shows that the WSN sensors are reconfigured as per the specifications and each neighborhood is covered by typically one active node while others remain in stand-by mode (energy of each note is decreasing during each step based on the model specification).

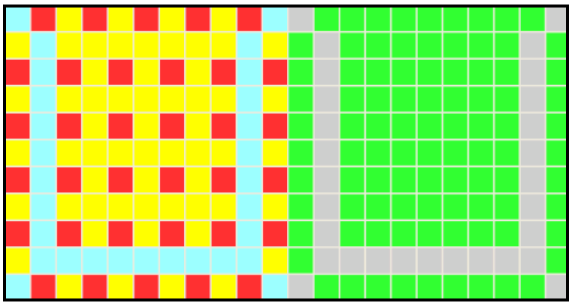


Fig.8 - Snapshot of simulations results after 15 time units

Fig.9 shows that several active nodes (from previous step) are now passive (died cells - Plane 1 on the right, gray cells), while cells that were previously in stand-by mode are taking over by becoming active.

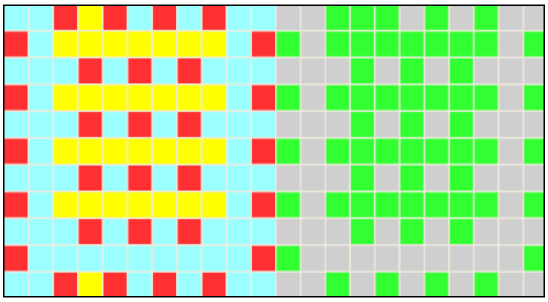


Fig.9 - Snapshot of simulations results after 62 time units

In these simulations, all the sensor nodes become passive after 197 time steps. By comparing the end result of our simulation and the results obtained in (Cunha, Silva, and Loreiro 2005) in particular, the number of active sensors after 200 time units), it can be observed that after approximately 200 time steps all the active sensors become passive (as energy of all nodes is consumed). Hence, the simulation results obtained by our model very closely reflect the same behavior (i.e. after 197 time units the active sensors become passive). These results can be seen in Fig.10.

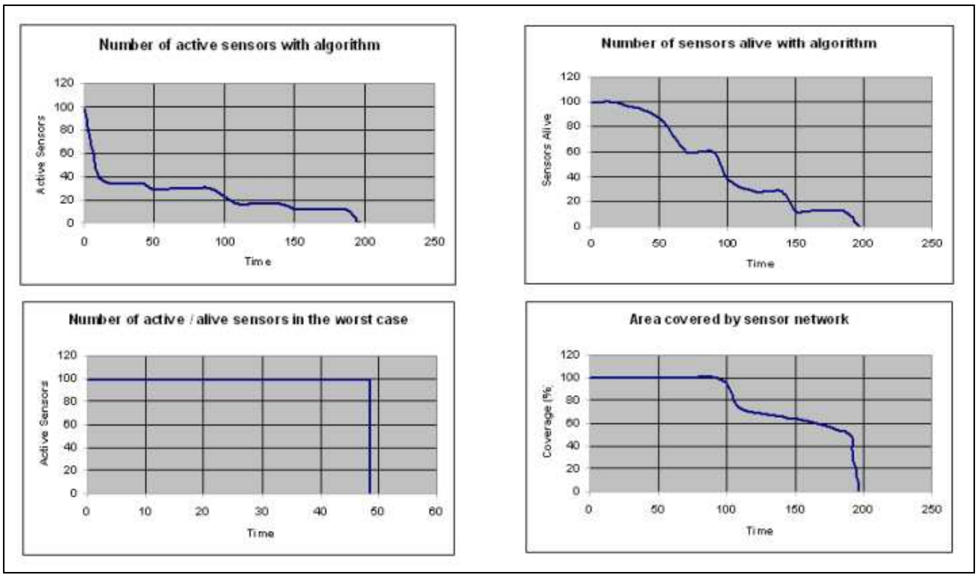
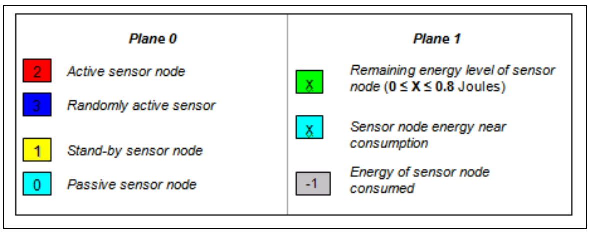


Fig.10 - Graphical representation of the Initial WSN simulation results using Cell-DEVS

From the Fig.10 it can be observed that in the worst case scenario (bottom left graph), when all nodes are active all the time, the sensor nodes die after 48.48 time units (where the energy of a node in each step is consumed by 0.0165 J and the initial level of sensors' energy is 0.8 J), while when using the Topology Control Algorithm, where selected sensor nodes within a Moore's neighborhood are alive (top left graph), the number of alive/active sensor nodes decreases gradually, extending the life of sensor nodes to 197 time units (top right graph) and the coverage area (bottom right graph).

Our following results present an improved version of the WSN model, in which we use the following is the graphical representation of the sensor node states defined within the simulation model:



Plane 0

2 Active sensor node

3 Randomly active sensor

1 Stand-by sensor node

0 Passive sensor node

Plane 1

x Remaining energy level of sensor node (**0 ≤ X ≤ 0.8** Joules)

x Sensor node energy near consumption

-1 Energy of sensor node consumed

Fig.11 - Possible sensor states of the improved WSN model

With respect to the initial model (refer to Fig.5), in representing the sensor node states of the improved model, we have added an addition sensor state to indicate the randomly active sensor nodes, as depicted in Fig.11.

This model provides closer approximation to the WSN topology algorithm by implementation of the randomness within the WSN deployed. The actual results obtained, more closely reflect the real-world scenarios and provide better insight into the WSN topology problem and how efficiently similar problems can be implemented utilizing Cell-DEVS and CD++ toolkit. The improved model is quite easy to modify in order to simulate different sizes of WSN networks. In this case, it requires to change only the values *n* and *m* (i.e., one line of specification, *dim : (n, m, 2)* where *n* is the number of columns and *m* is the number of rows in the model and provide the desired initial energy levels for the sensor nodes). The example below (refer to Fig.12) represent the simulation model for WSN33, where the cell space is constituted by 33 rows and 33 column, and the total number of cells (sensor nodes) is 1089.

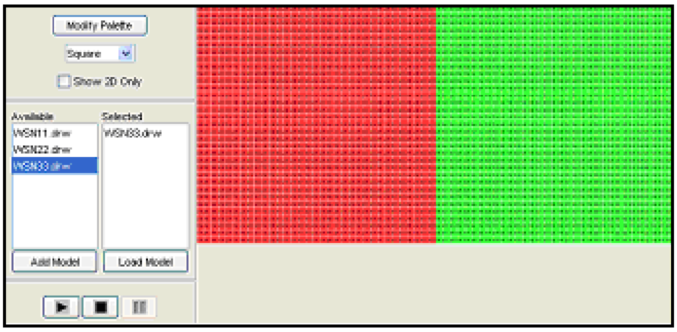


Fig.12 - Snapshot of Plane 0 (left, red) and Plane 1 (right, green) prior to execution

The following figure shows the 33 x 33 WSN network during the process of reconfiguration, where nodes are trying to form a structure and every neighbor is trying to set some sensors in active while others remain in stand-by mode. Some of the stand-by nodes are randomly awaken (in Fig.13 left, the cells in blue in Plane 0). In case that only one or two sensor nodes are active, stand-by nodes randomly become active and return to stand-by mode only if one or two more sensor nodes are currently active within the neighborhood. On the Plane 1 data (refer to Fig.13 right), we can see the energy level of each sensor node up to this time step (see in green, indicating that the energy levels x of sensor nodes are: $x \geq 0.05$J).

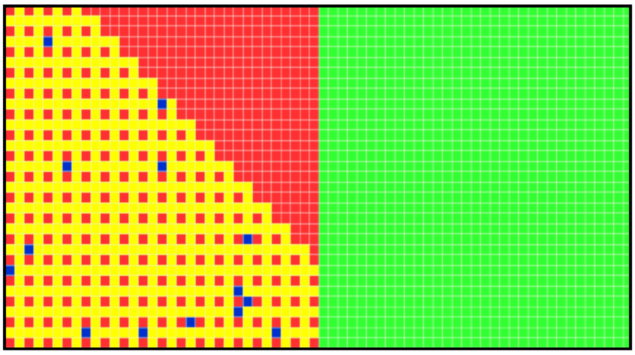


Fig.13 - Snapshot of simulations results after 40 time units

Studying the simulation results of this Cell-DEVS model (presented in Fig. 14 and Fig 15), we can be see that the coverage area by sensors is reduced after 134 time steps, when more and more nodes become passive (as their energy is consumed). As time progresses, there is a smaller area of the WSN cell-space covered. Finally, after 193 time units, the WSN cell-space becomes passive. Similar, results were obtained when WSN22 and WSN11 were simulated.

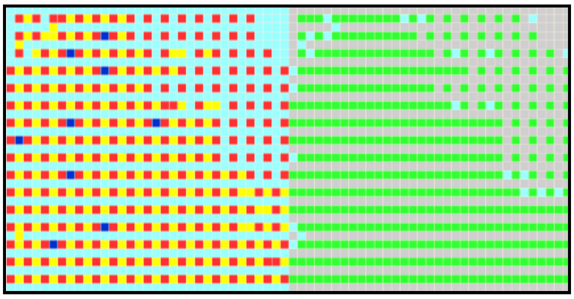


Fig.14 - Snapshot of simulations results after 134 time units

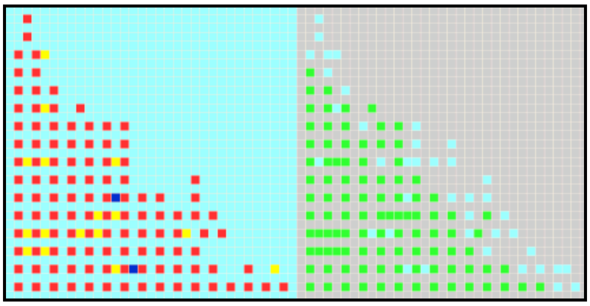


Fig.15 - Snapshot of simulations results after 164 time units

Fig.16 below shows the number of active sensors versus time within the WSN network; evaluated for WSN deployment scenarios within two dimensional cell-spaces, using Cell-DEVS:

a) WSN11 - representing *11x11* cell-space, with 121 sensor nodes
b) WSN22 – representing *22x22* cell-space, with 484 sensor nodes
c) WSN33 – representing *33x33* cell-space, with 1089 sensor nodes

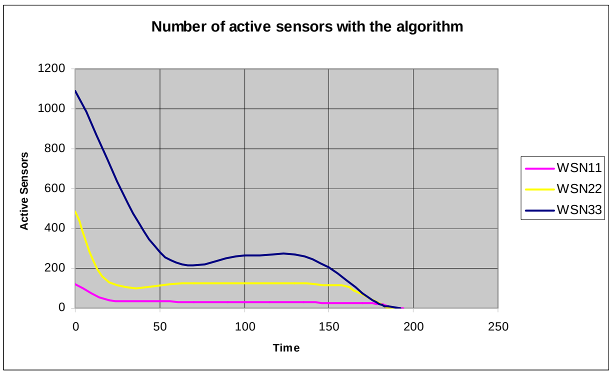


Fig.16 - Number of active sensors with the algorithm using Cell-DEVS

As we can see, the number of active sensors decreases after the configuration of sensor nodes, following the deployment, after which, redundancy is reduced by having only one active node within the Moore’s neighborhood (while other nodes are in stand-by). In addition, within each Moore’s neighborhood, the stand-by nodes become active randomly and remain active if no sensor is active or return to stand-by mode if any sensor node is still active. The results obtained provide clear indication that network lifetime is increase approximately by 4, which was shown also in (Cunha, Silva, and Loreiro 2005).

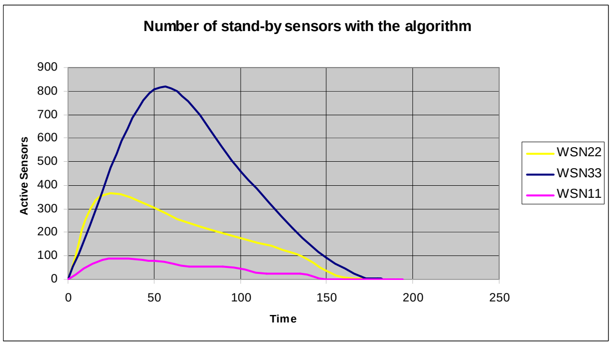


Fig.17 - Number of stand-by sensors with the algorithm using Cell-DEVS

On Fig.17, it can be observed that the number of stand-by sensors increases while the WSN cell-space is being configured, and starts to decrease as the active sensors’ energy is consumed (refer to Fig.16) hence stand-by sensors become active. Similarly, in Fig.18 we can be observe that the number of sensors alive is at it maximum when the simulation starts (i.e. all the WSN sensor nodes are active) and starts slowly decreasing as the time progresses.

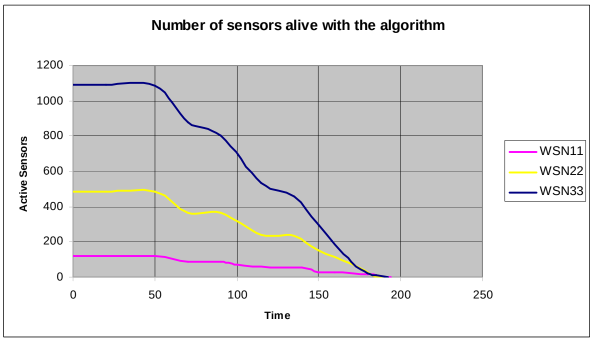


Fig.18 - Number of sensors alive with the algorithm using Cell-DEVS

An important aspect considered in WSN networks is the coverage area, which is closely related to the active sensors within the cell-space. When the redundancy of sensor nodes within the WSN is controlled, the network lifetime is prolonged; hence wider coverage area is maintained for a longer time, as shown by the simulation results in Fig.19.

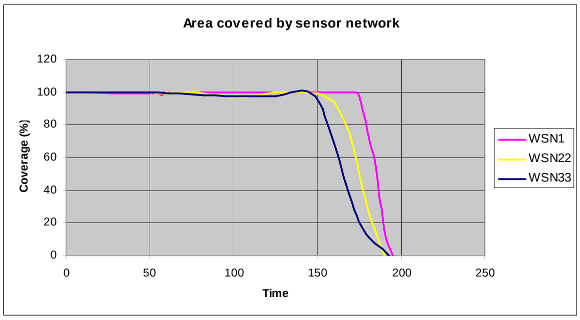


Fig.19 - Sensor network coverage area (%) using Cell-DEVS

Finally, in Fig.20 we can see the number of active sensors in the worst case scenario when WSN sensor nodes are active the entire time until the energy of sensors is consumed just before time 50; the network lifetime is approximately 4 time less in comparison to the implementation of WSN topology control algorithm using Cell-DEVS (refer to Fig.16, Fig.18 and Fig.19).

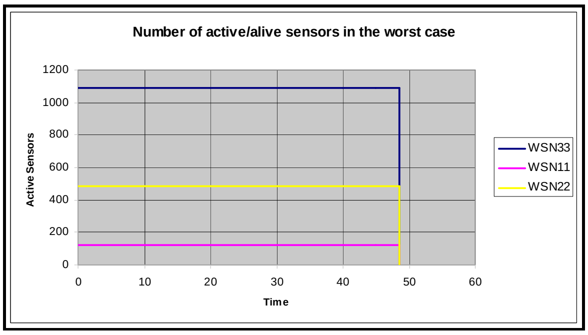


Fig.20 - Number of active sensors with the algorithm

4 CONCLUSIONS

The objective of this paper was to simulate a large Wireless Sensor Network (WSN) using Cell-DEVS, by implementing the Topology Control Algorithm as presented in. By observing and evaluating the behavior of WSN simulation model, under different test scenarios, it was proven the effectiveness of Cell-DEVS and the CD++ toolkit, as an elegant approach to model, simulate and analyze, in this case the WSN topology problem.

The initial WSN model was further enhanced in order to provide closer approximation to the WSN topology algorithm by implementation of the randomness within the WSN deployed. In addition, modification and simplification of model was done, where cell-space is not divided into zones; allowing more flexibility to model any possible WSN cell-space configuration (entire or partial cell-space populated with sensor node).

The actual results obtained with the improved model, more closely reflect the real-world scenarios and provide a better insight into the WSN topology problem, in particular. As it was observed in section 3, the complexity of the problem can be simplified and coded with less than 30 lines of code, utilizing CD++ toolkit and Cell-DEVS approach; whilst similar problem if implemented in C/C++ (or any other high level programming language) could possibly take up to several hundreds or perhaps thousands of lines of code (based on the implementation approach taken). Thus, proving how efficiently complex problems of similar nature can be implemented simulated and analyzed utilizing CD++ toolkit and Cell-DEVS approach.

ACKNOWLEDGMENTS

We would like to thank Energate team for their support during the preparation of this manuscript.

REFERENCES

Chopard, B. Droz, M. 1998. Cellular Automata Modeling of Physical Systems. *Cambridge Univ. Press*. New York, USA.

Cunha, R., A. Silva, and Loreiro, A. 2005. Simulating Large Wireless Sensor Networks Using Cellular Automata. *Proceeding of the 38th Annual Simulation* Symposium *ANSS'05*. IEEE Computer Society 2005, San Diego, CA, USA.

Healy, M. T.Newe, and E.Lewis. 2008. Wireless Sensor Node Hardware: A Review. *IEEE Sensors 2008 Conference*, Lecce, Italy.

Shen, Y, and B. Guo. 2007. Dynamic Power Management (DPM) based on Wavelet Neural Networks in Wireless Sensor Networks. *IFIP International Conference on Network and Parallel Computing*. Liaoning, China.

Wainer, G. 2009. Discrete-Event Modeling and Simulation: a practitioner's approach. CRC Press.

Wainer, G. 2002. CD++: a Toolkit to Define Discrete Event Models. *Software, Practice and Experience.* Wiley. Vol. 32, No. 3: 121-136.

Ye, F. G. Zhong, S.Lu, and L.Zhang. 2002. Energy Efficient Robust Sensing Coverage in Large Sensor Networks. Technical Report, UCLA Computer Science Department, CA, USA.

Zeigler, B. T. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems.* Academic Press.

BLERIM QELA (B.Eng., Electrical, M.Sc. Sys. Sci., P.Eng, received the B.Eng, Electrical Engineering in 2001. He has over 9 years of Engineering design experience. He worked as an Electronics Design Engineer for Tyco Electronics Ltd from 2001 to 2007, during which time he received a master's degree in Systems Science at the University of Ottawa (Jan. 2004 – July 2005). From 2007 to present, he works as an R& D Engineer for Energate Inc., and is also a graduate student of Electrical Engineering at the School of Information Technology and Engineering, University of Ottawa. His research interests are in WSN applications and Intelligent Systems.
Email: bqela@site.uottawa.ca

GABRIEL WAINER, SMSCS, SMIEEE, received the M.Sc. (1993) and Ph.D. degrees (1998, with highest honors) of the University of Buenos Aires, Argentina, and Université d'Aix-Marseille III, France. In July 2000, he joined the Department of Systems and Computer Engineering, Carleton University (Ottawa, ON, Canada), where he is now an Associate Professor. He has held positions at the Computer Science Department of the University of Buenos Aires, and visiting positions in numerous places, including the University of Arizona, LSIS (CNRS), University of Nice and INRIA Sophia-Antipolis (France). He is author of three books and over 190 research articles, edited four other books, and helped organizing over 90 conferences. He was PI of different research projects and recipient of various awards (NSERC, Precarn, Usenix, CFI, CONICET, ANPCYT, CANARIE, IBM Eclipse Innovation, SCS and others). He is the Special Issues Editor of the Transactions of the SCS, and Associate Editor of the International Journal of Simulation and Process Modeling. He was a member of the Board of Directors of the SCS, and a chair of the DEVS standardization study group (SISO). He is Director of the Ottawa Center of The McLeod Institute of Simulation Sciences and chair of the Ottawa M&SNet, and one of the investigators in Carleton University Centre for advanced Simulation and Visualization (V-Sim). His current research interests are related with modelling methodologies and tools, parallel/distributed simulation and real-time systems.
Email:gwainer@sce.carleton.ca; Website: www.sce.carleton.faculty/wainer/index.html

HUSSEIN MOUFTAH (PhD, FRSC, FCAE, FIEEE,FEIC) is a Canada Research Chair and Distinguished University Professor at the School of Information Technology and Engineering, University of Ottawa (Ottawa, ON, Canada) He has held positions in the Department of Electrical and Computer Engineering at Queen's University, BNR and Nortel Networks. Dr. Mouftah has published over 850 technical papers, 6 books and 30 book chapters. Dr. Mouftah has served the Institute of Electrical and Electronic Engineering (IEEE) Communications Society as Editor-in-Chief of the Communications Magazine (1995-97), Director of Magazines (1998-99), Chair of the Awards Committee (2002-03), Director of Education (2006-07), and Member of the Board of Governors (1997-99 and 2006-07). Also, he is the founding Chair of two of IEEE Communications Society's Technical Committees (TCs): Optical Networking TC (2002-04) and Ad Hoc and Sensor Networks TC (2005-07). He has been a Distinguished Speaker of the IEEE Communications Society (2000-07). Dr. Mouftah is the recipient of the 1989 Engineering Medal for Research and Development from the Association of Professional Engineers of Ontario and of the 2002 Ontario Distinguished Researcher Award of the Ontario Innovation Trust. He has also received 8 Out-

standing/Best Paper Awards (2 at ISCC’2008; ICC’2005; CITO Innovators Showcase’2004; IEEE Communications Magazine in 1993; SPECTS’2002; HPSR’2002; ISMVL’1984), the IEEE Canada Outstanding Service Award (1995), and the CSIM Distinguished Service Award of the IEEE Communications Society (2006). In 2004 Dr. Mouftah received the IEEE Communications Society Edwin Howard Armstrong Achievement Award and the George S. Glinski Award for Excellence in Research from the Faculty of Engineering, University of Ottawa. In 2006 he was honoured with the IEEE McNaughton Gold Medal and the Engineering Institute of Canada Julian Smith Medal. In 2007 he was the recipient of the Royal Society of Canada (RSC) Thomas W. Eadie Medal. Most recently, Dr. Mouftah received the University of Ottawa 2007-2008 Award for Excellence in Research and the ORION Leadership Award of Merit (2008). Dr. Mouftah is a Fellow of the IEEE (1990), Fellow of the Canadian Academy of Engineering (2003), Fellow of the Engineering Institute of Canada (2005) and Fellow RSC: The Academies of Canada (2008).
Email: mouftah@site.uottawa.ca;Website: www.site.uottawa.ca/~mouftah