HeliAmarasinghe 6152735(uOttawa)

Cell-DEVS Modeling and CD++ simulation of

Malware Propagation in Wireless Sensor Networks

Assignment 2

SYSC 5104 - Methodologies of Discrete Event Modeling and Simulations

Fall 2012

Carleton University

# Modeling and simulating malware propagation in sensor networks

## Introduction

Wireless sensor networks are widely used for monitoring applications in number of fields including health care, environment science, and chemical industries.

Wireless sensor networks differ from traditional computer networks in various aspects: First, WSNs are highly distributed system and consist of a great number of distributed nodes (sensor nodes) with the ability to monitor its surroundings. Second, sensor nodes are limited in power, computational capacities, and memory [1]. Finally, minimal (or no) human interaction for the sensors and self-organization is a fundamental feature of wireless sensor networks. Due to limitations of processing, memory and battery power, it is not always possible to execute strong malware resistance security algorithms in highly distributed sensor nodes.

In this work, we model and simulate malware propagation in distributed sensor networks when communication capabilities of each wireless node is limited by its battery power and wireless channel access is restricted by Media Access Control (MAC) layer rules.

## Conceptual model and description

We have modeled the wireless sensor network as a 2 dimensional lattice, with each cell may or may not occupied by a stationary wireless sensor node.



Each node can have following states;

|  |  |
| --- | --- |
| **Node States** | **Value** |
| Dead or unoccupied | -1 |
| Susceptible | 0 |
| Infected & spreading node | 1 |
| Infected & dormant node | 2 |

State transformation diagram for above states are as follows;



Initially all the nodes in the lattice are in the ***susceptible***state with fixed battery power. If no malware is received, it assumes sensor nodes perform their normal designed functions using relatively smaller battery power. Hence un-infected node will take long time to go to dead state. However if node receive a malware message, it moves to ***infected & spreading*** state and start broadcasting malware to its neighbors. Broadcasting consumes significant amount of power, draining its battery faster and result in moving to ***death*** state.

Wireless media access protocols are used in wireless sensor networks to avoid collision-free data transfer. We have used following media access states for each node to model wireless data transfer with minimal collisions.

|  |  |
| --- | --- |
| MAC layer node state | value |
| Free | 30 |
| Receiving | 31 |
| Broadcasting | 32 |

Due to media access (MAC layer) rules, node has to wait for channel to be free, to start broadcasting. Basic characteristic of wireless channel access is modeled by introducing ***infected & dormant*** state. Infected & spreading node wait for random delay before next malware broadcast by moving to infected & dormant state after each broadcast.

Finally sensor node battery power is modeled by starting with fixed sized battery power of value 20 and reduced to minimum of 10 in different rates for each node state.

## Cell-DEVS modeling

System is modeled in Cell-DEVS using three layers.



|  |  |  |
| --- | --- | --- |
| **Plane Name** | **Function** | **Specification** |
| Malware Plane (Pm) | Model malware infection | <Im, Xm, Ym, Xlistm, Ylistm, ηm, Nm, {rm, cm}, Cm, Bm, Zm, Selectm> |
| Power Plane (Pp) | Model battery usage | <Ip, Xp, Yp, Xlistp, Ylistp, ηp, Np, {rp, cp}, Cp, Bp, Zp, Selectp> |
| Access Plane (Pa) | Model media access | <Ia, Xa, Ya, Xlista, Ylista, ηa, Na, {ra, ca}, Ca, Ba, Za, Selecta> |

Cell-DEVS specifications for the modeled system are as follows;

**Malware Plane (Pm)**

**GCCbm = <Im, Xm, Ym, Xlistm, Ylistm, ηm, Nm, {rm, cm}, Cm, Bm, Zm, Selectm>**

**Xlistm**= { 0 };

**Ylistm**= { 0 }.

**ηm**= 6x6x3;

**Zm**:

[malware-propagation-rules]

rule : -1 1000 { (0,0,1)<11 }

rule : 1 1000 { ((0,0,0)=0 and (0,0,2)=31)}

rule : 2 1000 { (0,0,2)=32}

rule : 1 5000 { (0,0,0)=2}

rule : {(0,0,0)} 1000 {t}

**Nm**:

neighbors : sensor(-1,-1,0) sensor(-1,0,0) sensor(-1,1,0)

neighbors : sensor(0,-1,0) sensor(0,0,0) sensor(0,1,0)

neighbors : sensor(1,-1,0) sensor(1,0,0) sensor(1,1,0)

neighbors : sensor(0,0,1)

neighbors : sensor(0,0,2)

**Bm**= nowrapped;

**Sm**:

-1 : Dead node or Unoccupied by a sensor

0 : Susceptible node

1 : Infected & spreading

2 : Infected & dormant

**d** = inertial delay (default 1000)

**Power Plane (Pp)**

**GCCbp= <Ip, Xp, Yp, Xlistp, Ylistp, ηp, Np, {rp, cp}, Cp, Bp, Zp, Selectp>**

**Xlistp**= { 0 };

**Ylistp**= { 0 }.

**ηp**= 6x6x3;

**Zp**:

[power-usage-rules]

rule : 10 1000 { (0,0,-1)=-1}

rule : 10 1000 { (0,0,0)<=11 }

rule : {(0,0,0)\*.9} 1000 {(0,0,-1)=1}

rule : {(0,0,0)\*1.0} 1000 {(0,0,-1)=0 or (0,0,-1)=2 }

rule : {(0,0,0)} 1000 {t}

**Np**:

neighbors : sensor(0,0,-1)

neighbors : sensor(0,0,0)

**Bp**= nowrapped;

**Sp**:

Battery power

**d** = inertial delay (default 1000)

**Access Plane (Pa)**

**GCCbp= <Ia, Xa, Ya, Xlista, Ylista, ηa, Na, {ra, ca}, Ca, Ba, Za, Selecta>**

**Xlista**= { 0 };

**Ylista**= { 0 }.

**ηa**= 6x6x3;

**Za**:

[media-access-rules]

rule : 31 1000 { (0,0,0)=30 and statecount(32)>0}

rule : 30 1000 { (0,0,0)=31 and statecount(32)<1}

rule : 32 {round(uniform(1,10))\*100} { (0,0,-2)=1 and statecount(32)<1}

rule : 30 1000 { (0,0,0)=32}

rule : {(0,0,0)} 1000 {t}

**Na**:

neighbors : sensor(-1,-1,0) sensor(-1,0,0) sensor(-1,1,0)

neighbors : sensor(0,-1,0) sensor(0,0,0) sensor(0,1,0)

neighbors : sensor(1,-1,0) sensor(1,0,0) sensor(1,1,0)

neighbors : sensor(0,0,-1)

neighbors : sensor(0,0,-2)

**Ba**= nowrapped;

**Sa**:

30 : channel free

31 : receiving

32 : broadcasting

**d** = inertial delay (default 1000)

For this simulation, we have used a 6X6 lattice containing wireless sensor nodes. Also no special rule was defined for boarders since boarder cell behavior is similar to internal cells.Contents in the “SensorNetMalware.ma” file is as follows;

[top]

components: sensor

[sensor]

type: cell

dim: (6,6,3)

delay: inertial

defaultDelayTime: 1000

border:nowrapped

neighbors: sensor(-1,-1,0) sensor(-1,0,0) sensor(-1,1,0)

neighbors: sensor(0,-1,0) sensor(0,0,0) sensor(0,1,0)

neighbors: sensor(1,-1,0) sensor(1,0,0) sensor(1,1,0)

neighbors: sensor(0,0,1)

neighbors: sensor(0,0,-1)

neighbors: sensor(0,0,2)

neighbors: sensor(0,0,-2)

zone: media-access-rules { (0,0,2)..(5,5,2) }

zone: power-usage-rules { (0,0,1)..(5,5,1) }

localtransition: malware-propagation-rules

initialvalue: 0

initialCellsValue:sensorNet.val

[media-access-rules]

rule: 31 1000 { (0,0,0)=30 and statecount(32)>0}

rule: 30 1000 { (0,0,0)=31 and statecount(32)<1}

rule: 32 {round(uniform(1,10))\*100} { (0,0,-2)=1 and statecount(32)<1}

rule: 30 1000 { (0,0,0)=32}

rule: {(0,0,0)} 1000 {t}

[power-usage-rules]

rule: 10 1000 { (0,0,-1)=-1}

rule: 10 1000 { (0,0,0)<=11 }

rule: {(0,0,0)\*.9} 1000 {(0,0,-1)=1}

rule: {(0,0,0)\*1.0} 1000 {(0,0,-1)=0 or (0,0,-1)=2 }

rule: {(0,0,0)} 1000 {t}

[malware-propagation-rules]

rule: -1 1000 { (0,0,1)<11 }

rule: 1 1000 { ((0,0,0)=0 and (0,0,2)=31)}

rule: 2 1000 { (0,0,2)=32}

rule: 1 5000 { (0,0,0)=2}

rule: {(0,0,0)} 1000 {t}

**Descriptions of Rules**

[media-access-rules]

**rule : 31 1000 { (0,0,0)=30 and statecount(32)>0}**

If any of the neighbors are broadcasting=32, set state to receive=31

**rule : 30 1000 { (0,0,0)=31 and statecount(32)<1}**

If the the current state is receive and none of the neighbors are broadcasting, set state to free=30

**rule : 32 {round(uniform(1,10))\*100} { (0,0,-2)=1 and statecount(32)<1}**

If the node is infected=1 and media is free=30, start broadcasting=32

**rule : 30 1000 { (0,0,0)=32}**

After broadcasting, set media free=30

**rule : {(0,0,0)} 1000 {t}**

Default

[power-usage-rules]

Battery power of a node can vary from 10 to 20

**rule : 10 1000 { (0,0,-1)=-1}**

Set power value of all non-occupied cells to 10

**rule : 10 1000 { (0,0,0)<=11 }**

If the power value is less than 11, make it 10

**rule : {(0,0,0)\*.9} 1000 {(0,0,-1)=1}**

Power consumption rate of infected cells

**rule : {(0,0,0)\*1.0} 1000 {(0,0,-1)=0 or (0,0,-1)=2 }**

Power consumption rate of susceptible and recovered nodes

**rule : 10 1000 {t}**

Default

[malware-propagation-rules]

**rule : -1 1000 { (0,0,1)<11 }**

If battery power is less than 11, node will die

**rule : 1 1000 { ((0,0,0)=0 and (0,0,2)=31)}**

If susceptible node received malware message, go infected\_spread=1

**rule : 2 1000 { (0,0,2)=32}**

After doing a broadcast, go infected\_dormant=2

**rule : 1 5000 { (0,0,0)=2}**

After staying in infected\_dormant=2, go back to infected\_spread=1

**rule : {(0,0,0)} 1000 {t}**

Default

## Simulation Results

Cells in three planes are initialized using a .val file and output .log file is visualized using Drawlog tool and CD++ Modeler tool in the simulator. I this section, we have analyzedDrawlog outputs and corresponding CD++ Modeler outputs of first few important steps of the simulation execution.

Drawlog output

Line : 225 - Time: 00:00:00:000

**Malware-Plane Power-Plane Access-Plane**

**(0,0,0) to (5,5,0) (0,0,1) to (5,5,1) (0,0,2) to (5,5,2)**

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| 1 0 0 0 0 0| 0| 20 20 20 20 20 20| 0| 30 30 30 30 30 30|

1| 0 0 0 0 0 0| 1| 20 20 20 20 20 20| 1| 30 30 30 30 30 30|

2| 0 0 0 0 0 0| 2| 20 20 20 20 20 20| 2| 30 30 30 30 30 30|

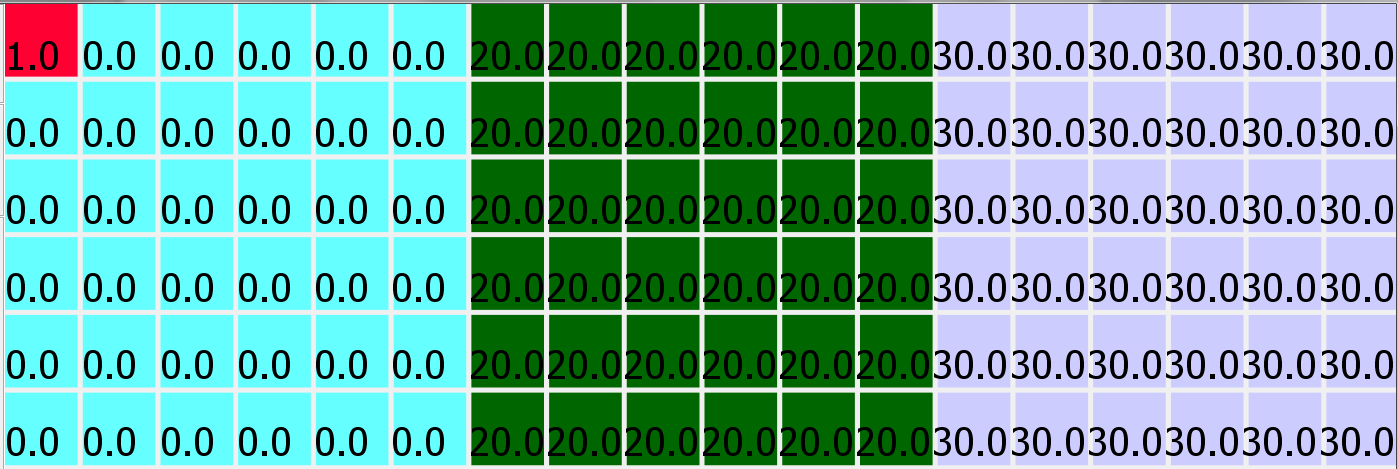
3| 0 0 0 0 0 0| 3| 20 20 20 20 20 20| 3| 30 30 30 30 30 30|

4| 0 0 0 0 0 0| 4| 20 20 20 20 20 20| 4| 30 30 30 30 30 30|

5| 0 0 0 0 0 0| 5| 20 20 20 20 20 20| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+

CD++ Modeler output



Initially the malware plane shows that all the nodes are insusceptible state except the node (0, 0, 0). We have initialized this node as an infected node to observe the malware propagation behavior. We have set the initial battery power level of every node to its maximum value, i.e. 20. The access plane is initialized to show that the wireless media is free and none of the nodes are broadcasting or receiving.

Line : 244 - Time: 00:00:00:600

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| 1 0 0 0 0 0| 0| 20 20 20 20 20 20| 0| 32 30 30 30 30 30|

1| 0 0 0 0 0 0| 1| 20 20 20 20 20 20| 1| 30 30 30 30 30 30|

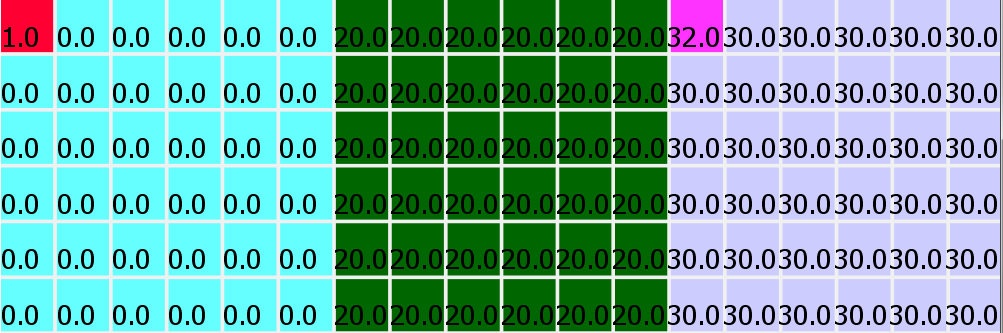
2| 0 0 0 0 0 0| 2| 20 20 20 20 20 20| 2| 30 30 30 30 30 30|

3| 0 0 0 0 0 0| 3| 20 20 20 20 20 20| 3| 30 30 30 30 30 30|

4| 0 0 0 0 0 0| 4| 20 20 20 20 20 20| 4| 30 30 30 30 30 30|

5| 0 0 0 0 0 0| 5| 20 20 20 20 20 20| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+



According to rules, infected node will start broadcasting after random countdown time. It can be noticed that in 600ms, node (0,0,0) started broadcasting according to the access plane.

Line : 322 - Time: 00:00:01:600

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| 2 0 0 0 0 0| 0| 18 20 20 20 20 20| 0| 30 31 30 30 30 30|

1| 0 0 0 0 0 0| 1| 20 20 20 20 20 20| 1| 31 31 30 30 30 30|

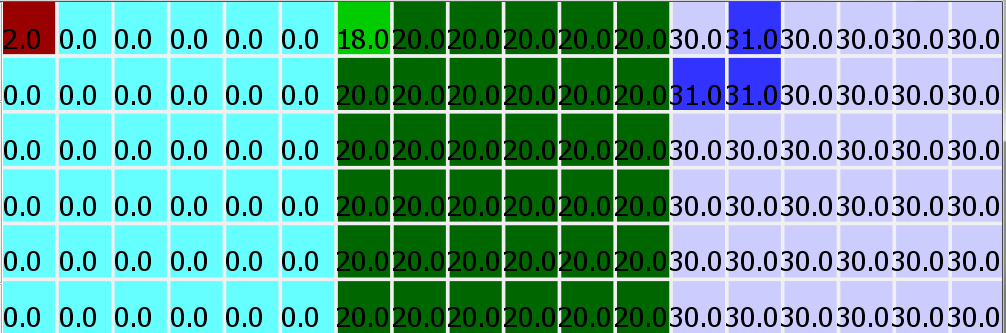
2| 0 0 0 0 0 0| 2| 20 20 20 20 20 20| 2| 30 30 30 30 30 30|

3| 0 0 0 0 0 0| 3| 20 20 20 20 20 20| 3| 30 30 30 30 30 30|

4| 0 0 0 0 0 0| 4| 20 20 20 20 20 20| 4| 30 30 30 30 30 30|

5| 0 0 0 0 0 0| 5| 20 20 20 20 20 20| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+



After 1000ms, node (0,0,0) has moved to infected & dormant state while its battery power has been reduced to 18. Also From access layer, it can be seen that the neighbouring nodes has moved to receiving state.

Line : 387 - Time: 00:00:02:600

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| 2 1 0 0 0 0| 0| 18 20 20 20 20 20| 0| 30 30 30 30 30 30|

1| 1 1 0 0 0 0| 1| 20 20 20 20 20 20| 1| 30 30 30 30 30 30|

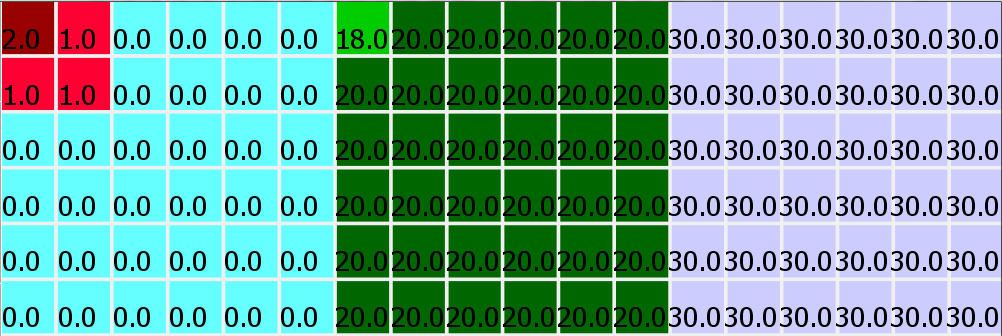
2| 0 0 0 0 0 0| 2| 20 20 20 20 20 20| 2| 30 30 30 30 30 30|

3| 0 0 0 0 0 0| 3| 20 20 20 20 20 20| 3| 30 30 30 30 30 30|

4| 0 0 0 0 0 0| 4| 20 20 20 20 20 20| 4| 30 30 30 30 30 30|

5| 0 0 0 0 0 0| 5| 20 20 20 20 20 20| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+



After 1000ms, three neighboring nodes of (0,0,0) has been infected by malware and at after random time, those neighboring nodes has started broadcasting as shown below.

Line : 430 - Time: 00:00:03:500

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| 2 1 0 0 0 0| 0| 18 20 20 20 20 20| 0| 30 32 30 30 30 30|

1| 1 1 0 0 0 0| 1| 20 20 20 20 20 20| 1| 32 32 30 30 30 30|

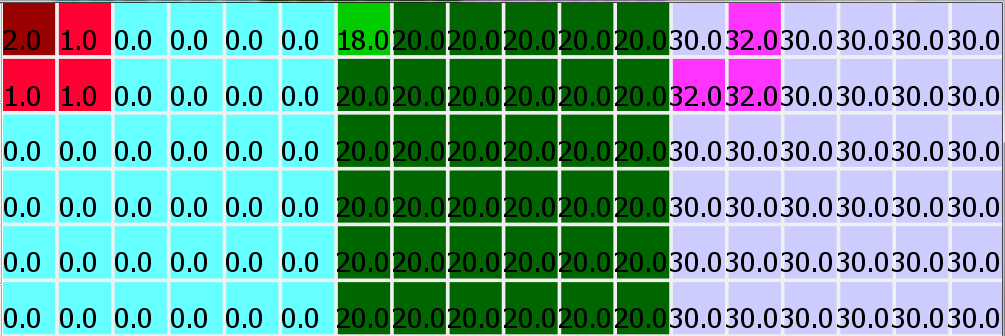
2| 0 0 0 0 0 0| 2| 20 20 20 20 20 20| 2| 30 30 30 30 30 30|

3| 0 0 0 0 0 0| 3| 20 20 20 20 20 20| 3| 30 30 30 30 30 30|

4| 0 0 0 0 0 0| 4| 20 20 20 20 20 20| 4| 30 30 30 30 30 30|

5| 0 0 0 0 0 0| 5| 20 20 20 20 20 20| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+



Neighboring susceptible nodes which receive these broadcasting messages gets infected and battery power of the nodes which broadcast the malware will be reduced. Infected and Dormant nodes will move back to infected and spreading state after waiting for a random time and will broadcast malware repeatedly until its battery drained out.

Line : 4735 - Time: 00:00:14:200

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| 1 2 1 1 2 2| 0| 15 16 15 18 18 18| 0| 30 30 31 31 31 30|

1| 1 2 -1 1 1 1| 1| 12 13 10 11 13 16| 1| 32 30 31 30 31 31|

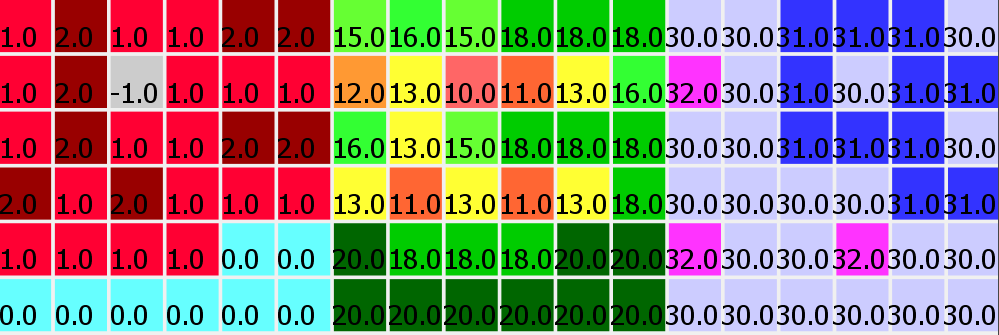
2| 1 2 1 1 2 2| 2| 16 13 15 18 18 18| 2| 30 30 31 31 31 30|

3| 2 1 2 1 1 1| 3| 13 11 13 11 13 18| 3| 30 30 30 30 31 31|

4| 1 1 1 1 0 0| 4| 20 18 18 18 20 20| 4| 32 30 30 32 30 30|

5| 0 0 0 0 0 0| 5| 20 20 20 20 20 20| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+



This intermediary state taken at simulation time 14000ms shows that after the battery power has been completely drained out, the moves to dead state. Finally the simulation stops when all the nodes are dead as shown below.

Line : 14586 - Time: 00:00:39:800

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5

+------------------+ +------------------+ +------------------+

0| -1 -1 -1 -1 -1 -1| 0| 10 10 10 10 10 10| 0| 30 30 30 30 30 30|

1| -1 -1 -1 -1 -1 -1| 1| 10 10 10 10 10 10| 1| 30 30 30 30 30 30|

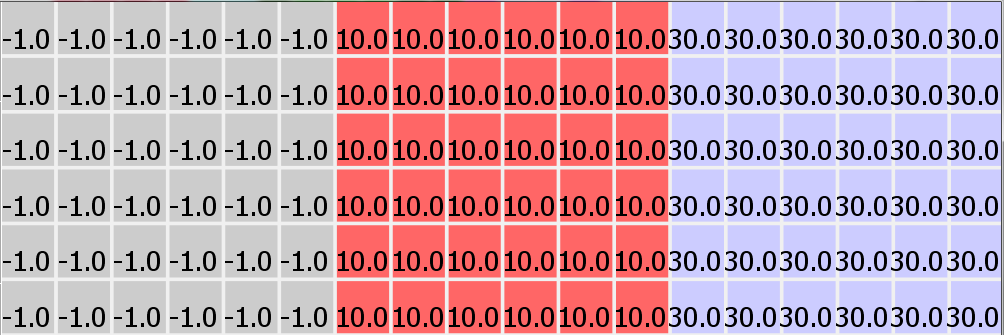
2| -1 -1 -1 -1 -1 -1| 2| 10 10 10 10 10 10| 2| 30 30 30 30 30 30|

3| -1 -1 -1 -1 -1 -1| 3| 10 10 10 10 10 10| 3| 30 30 30 30 30 30|

4| -1 -1 -1 -1 -1 -1| 4| 10 10 10 10 10 10| 4| 30 30 30 30 30 30|

5| -1 -1 -1 -1 -1 -1| 5| 10 10 10 10 10 10| 5| 30 30 30 30 30 30|

+------------------+ +------------------+ +------------------+



## References

[1] Y. Song and G.P. Jiang, “Modeling malware propagation in wireless sensor networks using cellular automata,” in IEEE international conference on Neural Networks and Signal Processing, Zhenjing, China, June 2008.

[2] CD++ User’s Guide, Daniel A. Rodriguez, Gabriel A. Wainer, Departamento de Computación, Universidad de Buenos Aires, Buenos Aires, Argentina, 1999