**SYSC 5104**

**Methodologies for Discrete Event Modelling and Simulation**

**ASSIGNMENT 2**

“Network Decontamination with Temporal Immunity by Cellular Automata”

**Submitted By: Submitted to:**

**Bhumika Patel (101047616) Prof. Gabriel Wainer**

**Sagar Patel (101053635)**

# **1.0) INTRODUCTION**

The objective of this work is to simulate a Network decontamination implementing the common network topology with a 2-dimensional 3-states automata with and without temporal immunity as presented in [1] using the Cell-DEVS model, which enables efficient execution of cellular automata models.

In our work we are going to consider the problem of network decontamination which is widely studied in distributed computing. Considering [1] we will describe the global disinfection process without using active agents and using the cell automata rules.

We will be considering a 2 dimensional cellular automata grid like in [1] using the disinfection rules and distinguishing between Von Neumann and Moore Neighbourhood that a bigger neighbourhood allows for a better disinfection.

Some computing nodes in the network may contain viruses and behave incorrectly, these are the contaminated nodes. These contaminated nodes propagates faulty computations and may also contaminate the other neighbouring nodes and cause the whole network functioning incorrect. But, due to the antivirus software running on some nodes, leaves it decontaminated for certain period. Nodes cannot detect the virus in them and in their neighbourhood so the spread of contamination is not controlled. The nodes can however, detects the process of antiviral is active or has been activated in their neighbourhood. When antivirus is active, it interrupts other local computations. In this work, we are simulating the strategy which is devised in [1] to make all nodes clean, regardless of the contamination pattern.

Here, to calculate the neighbourhood, we are using Von Neumann and Moore Neighbourhood having N neighbourhood and three possible states:

* 0 – Disinfected: Disinfecting node will leave the node *disinfected*, also the node which have the antiviral software is *disinfected node*.
* 1 – Disinfecting: The node running the antiviral software is *disinfecting node.*
* 2 – Unprotected: The node which never runs the antiviral software is the *unprotected node*.

A 2-dimensional cellular automata (CA) can be described by a quadruple C = <Z2, {0, 1, 2}, N, f> where: Z2 represents the set of cells (also called sites or nodes); {0, 1, 2} is the set of states of the cells; N is the neighbourhood of a cell, and f : {0, 1, 2} |N| → {0, 1, 2} is the local transition rule (or simply local rule) of the automaton.[1] The Von Neumann neighborhood contains the cell itself and the four cells at distance one in the Manhattan norm while the Moore neighborhood considers the cells of the Von Neumann neighborhood and also includes the four neighboring“diagonal cells”for given cell( i, j ).[1]

**Von-Neumann Neighbourhood:**

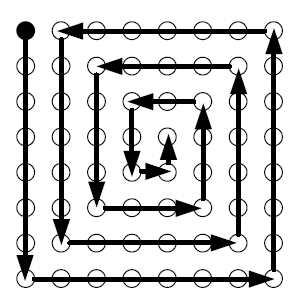
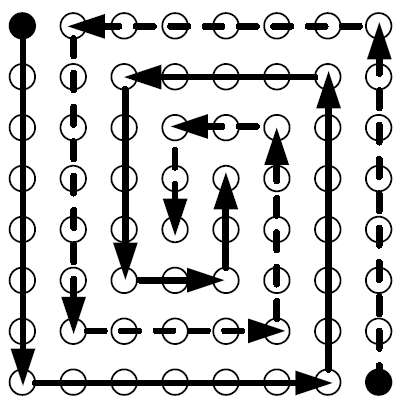
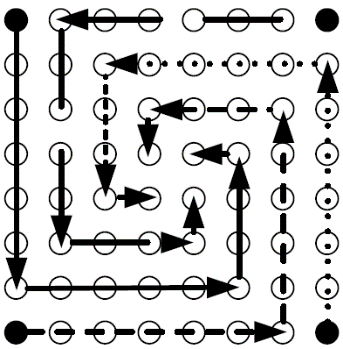
  

Fig.1 Propagation of disinfection with one, two and four disinfecting node using Von Neumann neighbourhood [1]

**Moore Neighbourhood:**

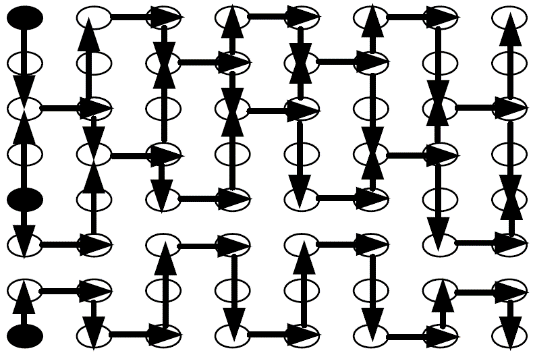


Fig.2 Propagation of disinfection with several disinfecting node using Moore neighbourhood [1]

Here, we are choosing the location of the initial disinfecting sites in such a way that during the evolution a disinfected node never comes into contact with an unprotected site. A disinfecting cell becomes disinfected and one or more of its neighbours become disinfecting, thus simulating the propagation of disinfection from cell to neighbouring cell(s).[1]

# **2.0) FORMAL SPECIFICATION FOR NETWORK DECONTAMINATION MODEL**

The following are the formal specification for the Cell-DEVS Network decontamination model:

**CD = < X, Y, I, S, θ, N, d, δint, δext, τ, λ, D >**

X =Y= {0, 1, 2}

S = {0, 1, 2}

N = neighborhood = { (-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1) }

d (transport delay) = 100 ms

τ = {

if (current cell == disinfecting)

{

delay;

current cell=disinfected;

}

if (cell(1,0) == unprotected)

{

Cell(1,0)=disinfecting;

}

else if (cell(0,1) == unprotected)

{

Cell(0,1)=disinfecting;

}

else if (cell(-1,0) == unprotected)

{

Cell(-1,0)=disinfecting;

}

else if (cell(0,-1) == unprotected)

{

Cell(0,-1)=disinfecting;

}

}

I, θ, δint, δext, λ, D defined according to Cell DEVS definitions.

# **3.0) RESULTS**

In the results, Yellow nodes shows Disinfecting nodes, Green nodes shows unprotected nodes and grey nodes shows disinfected nodes.

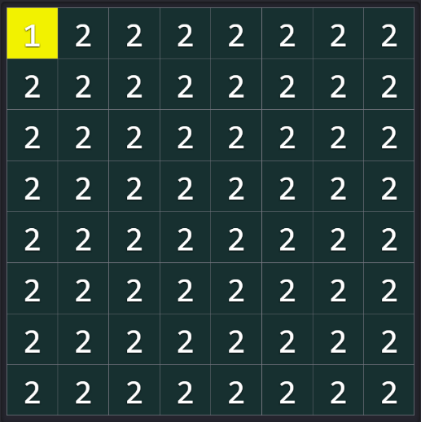
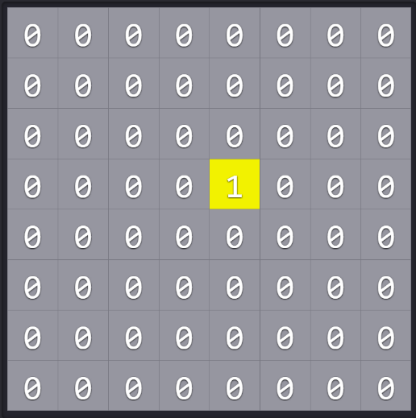
  

Fig.3 Initial and final phase of One Disinfecting nodes in network using Von Neumann Neighborhood

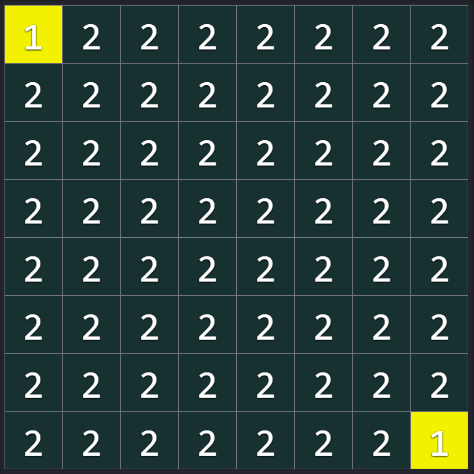
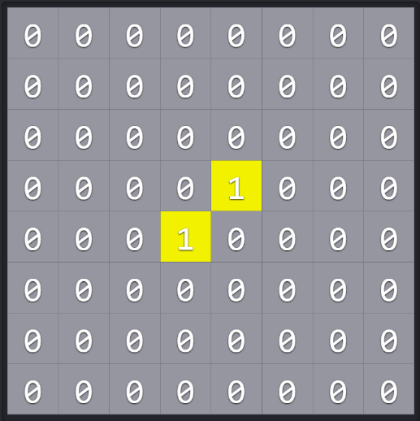
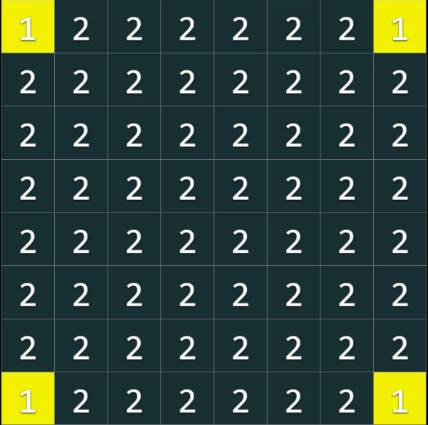
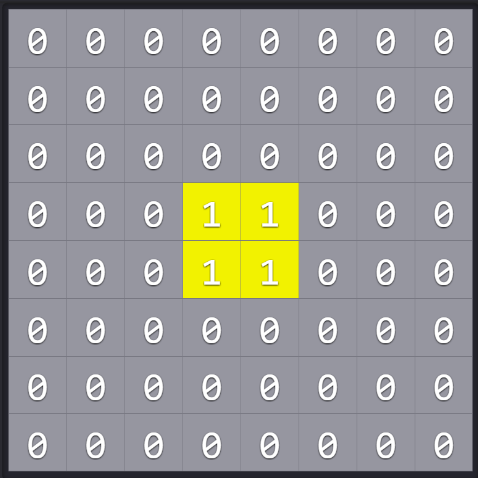
  

Fig.4 Initial and final phase of Two Disinfecting nodes in network using Von Neumann Neighborhood

  Fig.5 Initial and final phase of Four Disinfecting nodes in network using Von Neumann Neighborhood

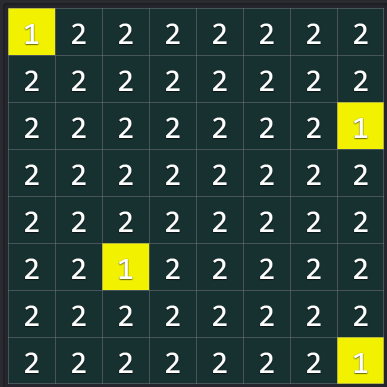
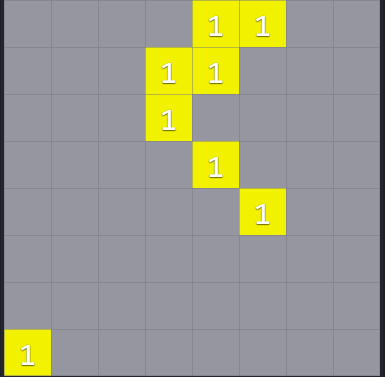
  

Fig.6 Initial and final phase of Four Disinfecting nodes in network using Moore Neighborhood

**Conclusion:**

After running the model, we got the expected results from network decontamination model in Cell-DEVS. The nodes are finding their neighbourhood by Moore’s neighbourhood and Van Neumann neighbourhood algorithms and it is performing the decontamination of network successfully as per the expected outcomes.

**References:**

[1] D. Yassine, F. Paola, Z. Nejib, “*Network Decontamination with Temporal Immunity by Cellular Automata*” In proceedings of 9th International Conference on Cellular Automata for Research and Industry, ACRI 2010, Ascoli Piceno, Italy, September 21-24, 2010.