

# CDA6530: Performance Models of Computers and Networks (Fall 2014)

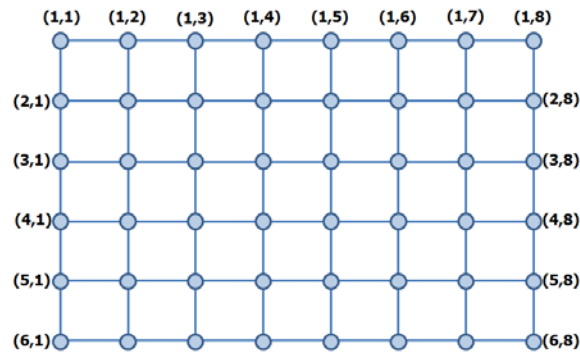
## Project 3: Topological Malware Propagation Discrete-time Simulation

(Assigned Oct. 13<sup>th</sup>; due: Oct. 26<sup>th</sup> midnight via webCourse)

You are required to simulate topological malware propagation in a simplified network by using discrete-time simulation technique. Until nowadays we still do not have a good analytical model to model malware spreading in a topological network. Email virus/worm, social-network based virus/worm, smartphone virus/worm based on Bluetooth communication, etc., all of them can be treated as topological malware.

### S1). A strict two-dimensional grid topology.

Considering a computer network follows a strictly defined two-dimensional grid topology like the following:



Each node represents a computer that can be compromised by the malware. A malware-infected node can only infect nodes directly connected with it. Each node is uniquely identified by its position in the grid. For example,  $Node(m,n)$  represents the node at the  $m$ -th row and  $n$ -th column in the grid. Suppose the network under study has  $m$  rows and  $n$  columns (i.e., node population is  $N = m \times n$ ).

**Infection process:** Once a node is infected at the discrete time  $t$ , it will send out infectious traffic once and only once to arrive at all its neighbors at the time tick  $t+xI$ , where  $x$  is a random variable following Poisson distribution with parameter  $\lambda$  ( $x$  could take values of  $0,1,2,3,\dots$ ). Then at the time  $t+xI$ , If the neighbor is already infected, such traffic will be ignored; if the neighbor is still vulnerable, it will be infected with the fixed probability  $p$  at the current discrete time  $t+xI$ .

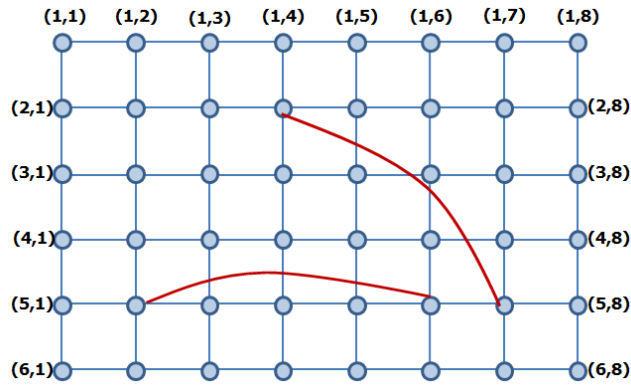
**Simulation:** Suppose the parameters are:  $m=300$ ,  $n=300$ ,  $\lambda=10$ ,  $p = 0.6$ . You need to conduct the simulation for 40 runs, in each simulation run the following 10 nodes are infected initially (i.e., at  $t=0$ ):  $Node(1,1)$ ,  $Node(2,2)$ ,  $Node(3,3), \dots, Node(10,10)$ ; and stop each simulation run when there will be no infectious traffic being generated any more. Obtain the total number of infected computers at each discrete time  $t$ , represented by  $I(t)$ . In the end, obtain the average value of  $I(t)$  at each discrete time  $t$  over those 40 runs.

### Report:

- (1). Draw a figure shows this average value of  $I(t)$  averaged over these 40 simulation runs.
- (2). Draw a figure shows the value of  $I(t)$  for the first 3 simulation runs.
- (3). Since we have obtained 40 values of  $I(t)$  for any given time  $t$  from those 40 simulation runs, what is the mean value and variance of this random variable  $I(t)$  when  $t$  goes to infinite? What is the mean and variance of  $I(t)$  when  $t=100$ ?

**S2). A small-world style two-dimensional grid topology.**

Now let us consider a more realistic grid topology in the following:



Besides the regular grid-type links among nodes, there are a few “shortcut” links connecting some pairs of nodes far away together. Such kind of topology is usually called “Small World” topology.

Now for the same simulation parameters:  $m=300$ ,  $n=300$ ,  $\lambda=10$ ,  $p = 0.6$ . Suppose the topology has the following 5 shortcut links:

(2,4) ----- (100, 250)                      (10, 150) ----- (250, 10)                      (100,100) ----- (200, 200)  
 (40, 100) ----- (100, 250)                      (20, 200) ----- (100,10)

Conduct the simulation 40 runs again, and provide report on the same 3 questions as in the **S1**.

**Submission:** Please submit your report document (a word file or PDF file), and your simulation codes. You can use Matlab or C or java or Python to program your simulation code. Your report should explain how you design your simulation, how I can compile/run and test your code, what important variables you used in your code. Explain the meaning of each figure you draw in your report document.

In addition, your result figures should be clearly readable in black-white printout (as I explained in class), which will be counted in your grade!