International Journal of Modern Physics C
Vol. 23, No. 10 (2012) 1250064 (16 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0129183112500647

A DISCRETE MATHEMATICAL MODEL TO SIMULATE MALWARE SPREADING

A. MARTIN DEL REY
E.P.S. de Ávila, Department of Applied Mathematics
Universidad de Salamanca
C/Hornos Caleros 50, 05003-Ávila, Spain
delrey@usal.es

G. RODRIGUEZ SÁNCHEZ
E.P.S. de Zamora, Department of Applied Mathematics
Universidad de Salamanca
Avda. Requejo 33, 49022-Zamora, Spain
gerardo@usal.es

Received 18 April 2012
Accepted 27 July 2012
Published 18 September 2012

With the advent and worldwide development of Internet, the study and control of malware spreading has become very important. In this sense, some mathematical models to simulate malware propagation have been proposed in the scientific literature, and usually they are based on differential equations exploiting the similarities with mathematical epidemiology. The great majority of these models study the behavior of a particular type of malware called computer worms; indeed, to the best of our knowledge, no model has been proposed to simulate the spreading of a computer virus (the traditional type of malware which differs from computer worms in several aspects). In this sense, the purpose of this work is to introduce a new mathematical model not based on continuous mathematics tools but on discrete ones, to analyze and study the epidemic behavior of computer virus. Specifically, cellular automata are used in order to design such model.

Keywords: Computer virus; network security; cellular automata; epidemic model.

PACS Nos.: 89.75.-k, 89.20.Ff, 89.20.Kk.

1. Introduction

Nowadays, the use of personal computers and the Internet (this fantastic, immense and global network that connects computers) makes it possible that people from all over the world can communicate and share information with little more than a few keystrokes. This stimulating environment also raises significant data security issues. In this sense malware attacks is the most common computer security problem and it is the source of the financial losses not only for companies and government agencies but also for people. The 2009 CSI Computer Crime and Security Survey (an annual report on the information and computer security of the USA corporations, government agencies, financial, educational and medical institutions and other organizations) states that malware infection continues to be the most commonly seen attack: Specifically, the 64.4% of security incidents was due to malware infections despite the fact that the 99.1% of the companies use anti-virus software, the 97.9% use firewalls and the 89.9% use anti-spyware software. Moreover, it estimates that the virus incidents costs is hardly a threat to the viability of most organizations.

Malware (short for malicious software) is a computer program designed specifically to damage or disrupt a computer system. As the first malware publicly appeared was computer viruses, the term "virus" has come to be used interchangeably with the generic term "malware" and specific types of such dangerous software as "worm." Nevertheless, computer virus is only a specific type of malware: Other categories include worms, Trojan horses, bots, logic bombs, spyware, rootkits, etc. (see Ref. 1). Consequently and unfortunately there is not a satisfactory definition of a computer virus because this notion has been overloaded with many definitions over the years, and, as is mentioned above, usually it has been mistaken for a Trojan horse or a worm (because of its similarities). Nevertheless, one can state that a computer virus is a hidden and malicious program that infects a computer by copying itself to other programs or files (usually executable program files, file directories, macros, system sector, etc.) The computer virus is executed (and runs secretly) when the host program or document is opened. Once the computer virus is initiated, it searches for uninfected files and tries to attach itself to them too. The action of computer viruses hinders the normal working of computers and includes deleting files, trashing the BIOS, leaving backdoors, spying private data, using the infected computer to mount DoS attacks, etc.

Consequently the design of mathematical models that allow one to simulate the computer virus spreading is an important issue. Several models have been published in the scientific literature to simulate the behavior of malware, and specially the spreading of computer worms in a computer network (see, for example, Refs. 2–7). Unfortunately, to the best of our knowledge, there is not any work dealing with virus computers instead of worm computers (a computer virus is a parasitic software whereas computer worms are not parasitic software). The majority of mathematical models to simulate malware spreading are based on the use of differential equations (see, for example, Ref. 1 and references therein). Sorrowfully, these models exhibit some important drawbacks since they do not take into account spatial factors such as population density, they neglect the local character of the spreading process, they do not include variable susceptibility of computers host, they cannot comprehensively depict complex contagion patterns (which are mostly caused by the human interaction induced by storage devices), etc. In this sense it is very interesting to study the use of discrete models (agent-based models, cellular automata (CA), etc.): They can eliminate the shortcomings exhibited by the models based on ODEs, and they are also specially suitable for computer simulations (see Ref. 8). Unluckily, discrete models have not been used for this purpose and the only two works found deal with worms (see Refs. 9 and 10). Consequently it is interesting to explore the possibilities of discrete models (specially CA) in this field and that is precisely the main goal of this work: To introduce a new and simple mathematical model to simulate computer virus spreading based on CA on graphs.

CA are finite state machines formed by a collection of n memory units called cells. At each time step, they are endowed with a state from the state set given by a finite field (see, for example, Refs. 11 and 12). The state of a particular cell is updated synchronously according to a specified rule function, whose variables are the states of the neighbor cells at the previous time step. This new approach allows one to simulate the behavior either local (each host computer) or global (the whole network of computers) by means of simple rules of transition that involve specific parameters related with the dynamics of the computer virus.

The rest of the paper is organized as follows: Section 2 is about the basic theory of computer viruses; In Sec. 3 the basic theory of CA on graphs is introduced; the mathematical model to simulate computer virus spreading is presented in Sec. 4; in Sec. 5 the analysis of the proposed model is given, and finally, in Sec. 6 the conclusions and further work is shown.

2. Computer Viruses

2.1. *The life cycle of a computer virus*

As is mentioned above, computer virus is a malware that, when executed, attacks the host (it may begin to damage the computer and its data: Corrupt programs or data files, destroy files, alter the system files, etc.) and tries to replicate itself into other executable code. The self-replication into existing executable code is the main characteristic of computer virus, and it is the basic difference between computer virus and computer worms (a computer worm is not a parasitic software: It is standalone and do not rely on other executable code; moreover, computer worms are able to spread by itself from machine to machine). The term of "computer virus" came about because of its similarities with biological viruses that require a host organism to live and reproduce (see Refs. 3−16). Depending on its malicious activity, the computer virus can be classified into several types: boot- (system-) sector viruses, file viruses, memory-resident viruses, polymorphic viruses, stealth viruses, multipartite viruses, macro-viruses, etc. (see Ref. 17).

The evolution of a computer virus from its creation to its (possible) eradication (in the host which has been infected) consists of the following stages (see Fig. 1):

(1) **Creation stage:** The computer virus is created by the programmer-author. Currently, in spite of the fact that many people think, it is easy to write a computer virus by using software programs called "Virus creators" which have been available on the Web for download. Consequently these virus creation

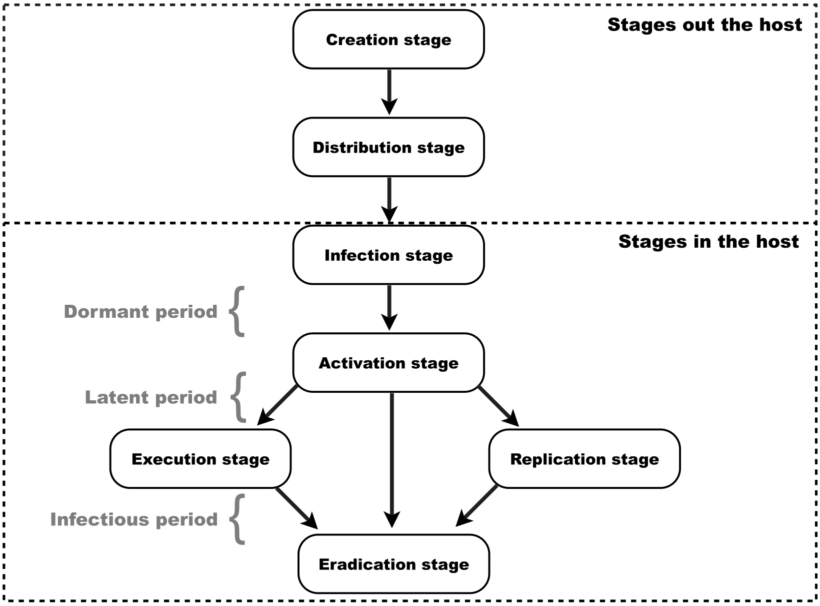


Fig. 1. Stages of the evolution of a computer virus.

utilities enable people with little or no programming knowledge to create a computer virus that can be very dangerous.

(2) **Distribution stage:** Once the computer virus has been developed, it must be hosted in an executable software to be distributed. This distribution can be done by storage devices although the most common vehicle for infection today is Internet and email.

(3) **Infection stage:** The computer is infected by acquiring the computer virus. In the early days of personal computers (when Internet was not used) computer virus were commonly distributed by human-transported media (floppy disk); currently, computer virus can be propagated by two possible ways:

- Physical devices: Floppy disks, CD-ROM, DVD-ROM, USB flash drive, external hard drive, memory cards, etc.
- Internet: With the evolution of Internet, downloaded files and email attached files are now the preferred delivery mechanism.

At this moment the computer virus stays hidden in the host program, remaining dormant until the host program will be run.

(4) **Activation stage:** When the host program is run, the computer virus runs secretly. At this moment the virus is triggered to perform its end goal: It waits for a pre-determined trigger (a specific date, a certain number of times the hosted program is executed, etc.) before delivering its "payload." The time between the infection stage and the activation stage is called dormant period.

(5) **Execution stage:** The virus performs its end goal. Payloads range from basic harmless messages to destruction of the core operating system. Sometimes the computer virus is not able to do its "dirty" work: It is present in the computer but not yet infecting anything or attacking; for example, a Windows virus can reside on a Linux-based machine and have no effect there, but can be exported to Windows machines. The time between the activation stage and the execution stage is called the latent period.
(6) **Replication stage:** The virus infects some area of the computer reproducing itself onto other programs (executable program files, file directories, macros, system sector, etc.) using the initial infected program to do so. Sometimes the virus fails to replicate which may occur as a result of bugs in the virus. The period in which the computer virus is executed and replicated is called infectious period.
(7) **Eradication stage:** It is the last stage of the life cycle of the virus. Once the computer virus has been detected by the user or the antivirus software, it must be removed from the computer: The infected programs and attachments must be cleaned of the virus. When it is not possible, the infected files must be deleted or put into "quarantine" where they cannot be run.

2.2. *Detection of a computer virus*

The discovery of a computer virus can be done by either the user or the protection software installed in the computer. Usually the users begin notifying system managers and other responsible parties of anomalous behavior in their systems (crashes, reboots, missing files, inoperability, etc.); moreover, the protection software (antivirus, firewalls, software for blocking) can discover the virus by examining files as they turn up at the computer and by scanning later at configurable pre-set times. The antivirus programs work by looking at each email attachment and downloaded file looking for virus "signatures." Some more sophisticated software exists for the purpose of blocking suspicious behaviors: Internet filters, email virus blockers, firewalls, etc.

In some cases, detection occurs early during the infection stage by means of the action of firewalls and software for blocking. In this case, the computer virus cannot reach the computer. During the dormant period and the latent period the computer virus can be discovered when the antivirus perform a routine scan of the system for virus signatures. In the infectious period both the user and the protection software installed in the computer can easily discover the virus (usually during this period its activity can be readily visible). Consequently the probability to be detected increases with the life cycle of the computer virus.

Once the virus is detected, it must be eliminated. Generally, the antivirus adopts one of two methods to eliminate the virus:

(1) **Removing the virus:** When the virus can be easily identified and can be removed without affecting other files, then the antivirus removes it from the host place.

(2) **Quarantine:** This is done when the virus cannot be easily identified and removed from the file (the removal of virus means the removal of the complete file). In this method, although the virus is not eliminated, it is rendered inactive by moving the file into "quarantine" and renaming it.

3. CA on Graphs

A graph G is a pair (V, E) where $V = \{v_1, v_2, \ldots, v_n\}$ is an ordered non-empty finite set of elements called nodes (or vertices), and E is a finite family of pairs of elements of V called edges. Two nodes of the graph, $v_i, v_j \in V$, are said to be adjacent (or neighbors) if there exists an edge in E of the form (v_i, v_j). We consider undirected graphs, that is, $(v_i, v_j) = (v_j, v_i) \in E$. A graph G is called simple if there is not two edges of G with the same ends and no loops exist, i.e. edges whose start and end is located at the same node.

If $V = \{v_1, \ldots, v_n\}$, the adjacency matrix of G is the $n \times n$ matrix, $A = (a_{ij})_{1 \leq i,j \leq n}$, where

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{if } (v_i, v_j) \notin E. \end{cases}$$

As this work deals with undirected graphs, the adjacency matrix is symmetric.

The neighborhood of a node $v \in V$, N_v, is the set of all nodes of G which are adjacent to v, that is: $N_v = \{u \in V \text{ such that } (v, u) \in E\}$. The degree of a node v, d_v, is the number of its neighbors.

A cellular automaton on an undirected graph $G = (V, E)$ is a 4-tuple $\mathcal{A} = (V, T, N, f)$ where: The set V defines the cellular space of the CA such that each node stands for a cell the cellular automaton. T is the finite set of states that can be assumed by the nodes at each step of time. The state of the node v at time step t is denoted by $s_v^t \in T$. These states change accordingly to the local transition function f. N is the neighborhood function which assigns to each node its neighborhood, that is:

$$\begin{aligned} N : V &\to 2^V, \\ v_i &\mapsto N(v_i) = N_{v_i} = \{v_{i_1}, v_{i_2}, \ldots, v_{d_i}\}, \end{aligned}$$

where, for the sake of simplicity, $v_{d_i} = v_{i_{d_{v_i}}}$. Note that the neighborhoods of the nodes are, in general, different from others. The local transition function f calculates the state of every node at a particular time step $t + 1$ from the states of its neighbors at the previous time step t, that is:

$$s_{v_i}^{t+1} = f\left(s_{v_{i_1}}^t, s_{v_{i_2}}^t, \ldots, s_{v_{i_{d_i}}}^t\right) \in T,$$

where $N_{v_i} = \{v_{i_1}, v_{i_2}, \ldots, v_{i_{d_i}}\}$.

This work deals with probabilistic CA, that is, those CA with a local transition function f involving some probabilistic parameters (the output of the local transition function could be different using the same input parameters).

4. The SEIQS Mathematical Model for Computer Virus Spreading

The model for computer virus spreading introduced in this work is a compartmental model, that is, each node/computer of the network can be at one of the following four states: S (susceptible), E (exposed), Q (quarantined) or I (infectious). Susceptible computers are those that have not been infected by the computer virus. A computer is exposed when it has been infected by the virus but it is non-activated (that is, they are the computer hosts during the dormant and latent period: The computer virus is in the infection stage or the activation stage). Infectious computers are those in which the computer virus is activated (execution stage) and it is able to propagate to another computer or file (replication stage). Infected computers are both exposed and infectious computers. Finally quarantined computers are those that have been detected as infected by the computer virus and have been removed to be cleaned (eradication stage). In Fig. 2 a flow diagram with the evolution of the states of a computer is shown.

In the model proposed in this work, the following assumptions will be done:

- The computer network is modeled as a graph with n nodes: $v_1, \ldots, v_n$. Each node stands for a computer and there is only one user associated to the node/computer: P_i is the unique user of computer v_i, $1 \leq i \leq n$.
- There is an edge between two nodes in the network if there is a relationship between theirs users that allows them to share software by physical contact or by email.
- Any node/computer is susceptible to be infected by the computer virus.
- The number of nodes in the network remains constant throughout time; at a particular time each node will be endowed with one of the following states: susceptible, exposed, infectious or quarantined.

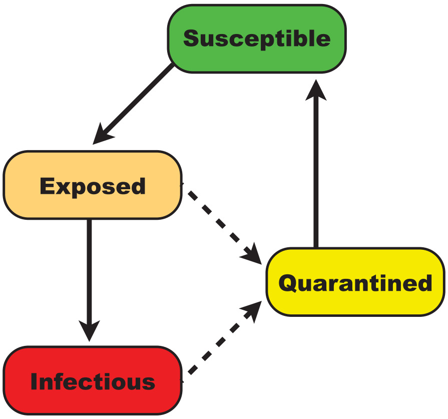


Fig. 2. (Color online) States of the computer where the computer virus is hosted.

The state of the node v_i at time step t is given by the binary vector $s_i^t = (S_i^t, E_i^t, I_i^t, Q_i^t) \in \mathbb{F}_2^4$, and consequently, the state set is

$$T = \{(1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1)\},$$

in such a way that if the node/computer v_i is susceptible then $s_i^t = (1,0,0,0)$; if it is exposed then $s_i^t = (0,1,0,0)$; if it is infectious then $s_i^t = (0,0,1,0)$, and finally, if the computer is quarantined $s_i^t = (0,0,0,1)$. The transition between these states is governed by means of local transition functions that, in our case, are boolean functions involving some probabilistic parameters. We will consider different transition functions depending on the initial and final states: From susceptible to exposed, from exposed to infectious or to quarantined, and from infectious to quarantined. In what follows these transition functions are detailed.

- **Transition from susceptible to exposed:** Taking into account Fig. 2 a susceptible computer v_i becomes exposed when the computer virus reaches it. As is mentioned in Sec. 2.1 the computer acquires the computer virus when: (1) Its user P_i has shared an infected software; (2) The user has downloaded an infected software from a web page; or (3) The user P_i opens an attached infected file from an email. Each of these three ways of infection is modeled as follows:

  (1) Let α_i^t be the probability to insert in the computer an external device and copy an infected file at time t; then we can define the boolean variable:

  $$X_i^t = \begin{cases} 1, & \text{with probability } \alpha_i^t, \\ 0, & \text{with probability } 1-\alpha_i^t. \end{cases}$$

  (2) Set β_i^t the probability (at time t) to visit a web page for downloading a file which could be suspected to be infected: Movies, computer games, cracked (owner) software, etc. Let pF_i^t be the probability that the firewall or the blocked software installed in the computer detects the infected file at time t. Then we can define the following boolean variable:

  $$Y_i^t = \begin{cases} 1, & \text{with probability } \beta_i^t \cdot (1-pF_i^t), \\ 0, & \text{with probability } 1-\beta_i^t \cdot (1-pF_i^t). \end{cases}$$

  (3) Let pV_{ij}^t be the probability to receive an email from the neighbor computer $v_j \in N_{v_i}$ at time t; set δ_i^t be the probability that the user P_i opens the attached file of an email at time t and γ_i^t the probability that the attached file will be infected. Then, for each neighbor node v_j of v_i, we consider the following boolean variable Z_{ij}^t:

  $$Z_{ij}^t = \begin{cases} 1, & \text{with probability } pV_{ij}^t \cdot \gamma_i^t \cdot \delta_i^t(1-pF_i^t), \\ 0, & \text{with probability } 1-pV_{ij}^t \cdot \gamma_i^t \cdot \delta_i^t \cdot (1-pF_i^t). \end{cases}$$

Consequently, the transition function that rules the evolution from susceptible to exposed is:

$$E_i^t = X_i^t \vee \left[c_i^t \wedge \left(Y_i^t \vee \bigvee_{v_j \in N_{v_i}} \left(I_{ij}^{t-1} \wedge Z_{ij}^t \right) \right) \right],$$

where c_i^t is the communication link parameter at time t associated to node v_i:

$$c_i^t = \begin{cases} 0, & \text{if the computer } v_i \text{ has no access to Internet at time } t, \\ 1, & \text{if the computer } v_i \text{ has access to Internet at time } t. \end{cases}$$

- **Transition from exposed to infectious:** Once a computer has been reached by the virus, it becomes infectious after a period of time which is computed as the sum of the dormant period (tD_i) and the latent period (tL_i). Consequently,

$$I_i^t = \begin{cases} 1, & \text{if } E_i^{t-1} = 1 \text{ and the } tE_i^{t-1} = tD_i + tL_i \\ 0, & \text{if } E_i^{t-1} = 1 \text{ and the } tE_i^{t-1} < tD_i + tL_i, \end{cases}$$

where the parameter tE_i^t stands for the discrete steps of time passed from the acquisition of the computer virus.

- **Transition from exposed to quarantined:** The computer virus hosted in an exposed computer can be detected by the antivirus software with a different probability depending on the period of time in which the computer virus is (dormant or latent period). As a consequence, if pD_i^t is the probability to be detected at dormant period and pL_i^t is the probability to be detected at latent period $(pD_i^t \leq pL_i^t)$, then:

$$Q_i^t = \begin{cases} 1, & \text{with probability } pD_i^t \text{ if } E_i^{t-1} = 1 \text{ and } tE_i^t \leq tD_i, \\ 1, & \text{with probability } pL_i^t \text{ if } E_i^{t-1} = 1 \text{ and } tD_i < tE_i^t \leq tD_i + tL_i, \\ 0, & \text{with probability } 1 - pD_i^t \text{ if } E_i^{t-1} = 1 \text{ and } tE_i^t \leq tD_i, \\ 0, & \text{with probability } 1 - pL_i^t \text{ if } E_i^{t-1} = 1 \text{ and } tD_i < tE_i^t \leq tD_i + tL_i. \end{cases}$$

- **Transition from infectious to quarantined:** During the infected period and for a computer virus, let puI_i^t be the probability to be detected by the user P_i at time t and set pI_i^t the probability to be detected by the antivirus (note that $pD_i^t \leq pL_i^t \leq pI_i^t$). Then if $I_i^{t-1} = 1$, it is,

$$Q_i^t = \begin{cases} 1, & \text{with probability } \max\left(puI_i^t, pI_i^t\right), \\ 0, & \text{with probability } 1 - \max\left(puI_i^t, pI_i^t\right). \end{cases}$$

- **Transition from quarantined to susceptible:** Finally, let tR_i be the period of time needed for a quarantined computer v_i to be cleaned from the computer virus. Then:

$$S_i^t = \begin{cases} 1, & \text{if } Q_i^{t-1} = 1 \text{ and the } tQ_i^{t-1} = tR_i, \\ 0, & \text{if } Q_i^{t-1} = 1 \text{ and the } tQ_i^{t-1} < tR_i, \end{cases}$$

where the parameter tQ_i^t stands for the discrete steps of time passed from the acquisition of the quarantined state by the computer v_i.

5. Examples and Discussion

In this section some simulations taking into account different scenarios will be done and analyzed. There are several parameters involved in the model which are summarized in Table 1.

In the simulations the following assumptions will be made:

- With respect to the probability to detect the computer virus:

$$pF_i^t \simeq pD_i^t \leq pL_i^t \leq pI_i^t.$$

- With respect to the length of the different periods:

$$1 \leq tD_i \leq 2, \quad 2 \leq tL_i \leq 4, \quad 1 \leq tR_i \leq 2.$$

- With respect to the users we classify them into four categories depending on their security abilities: Experienced users (class A), ordinary users awareness of security (class B), ordinary users not awareness of security issues (class C) and novice users (class D). The value of the parameters for each category is given in Table 2.
- The connection factor, c_i^t, follows a Bernoulli distribution with parameter 0.95, that is, $c_i^t = 1$ with probability 0.95 and $c_i^t = 0$ with probability 0.05.
- The probability that an attached file to an email is infected is considered small, that is: $\gamma_i^t \simeq 0.01$.

Table 1. Parameters involved in the model.

Parameter	Description
α_i^t	Probability to acquire the virus from an external device
β_i^t	Probability to download an infected file from a web page
pF_i^t	Probability to detect the computer virus by means of security software
pV_{ij}^t	Probability to receive an email from the computer v_j
δ_i^t	Probability to open an attached file of an email
γ_i^t	Probability that the attached file to an email will be infected
c_i^t	Communication link parameter to Internet
tD_i	Dormant period
tL_i	Latent period
tE_i^t	Time passed from the acquisition of the virus
pD_i^t	Probability to detect the computer virus during the dormant period
pL_i^t	Probability to detect the computer virus during the latent period
puI_i^t	Probability to detect the computer virus by the user during the infectious period
pI_i^t	Probability to detect the computer virus during the infectious period by means of antivirus
tR_i	Time needed to clean the computer v_i
tQ_i^t	Time passed from the computer was put in quarantine

Table 2. The different classes of users and their associated parameters.

Type	α_i^t	β_i^t	δ_i^t	pF_i^t	pD_i^t	pL_i^t	pI_i^t	puI_i^t
A	$[0, 0.1]$	$[0.1, 0.2]$	$[0.25, 0.45]$	$[0.5, 0.75]$	$[0.5, 0.6]$	$[0.6, 0.7]$	$[0.7, 0.8]$	$[0.8, 0.9]$
B	$[0.2, 0.3]$	$[0.2, 0.4]$	$[0.5, 0.7]$	$[0.3, 0.5]$	$[0.3, 0.4]$	$[0.5, 0.6]$	$[0.7, 0.8]$	$[0.7, 0.8]$
C	$[0.3, 0.4]$	$[0.4, 0.6]$	$[0.7, 1]$	$[0.1, 0.3]$	$[0.2, 0.3]$	$[0.4, 0.5]$	$[0.6, 0.7]$	$[0.5, 0.7]$
D	$[0.4, 0.5]$	$[0.6, 0.8]$	1	0	$[0.1, 0.2]$	$[0.2, 0.3]$	$[0.3, 0.4]$	$[0, 0.4]$

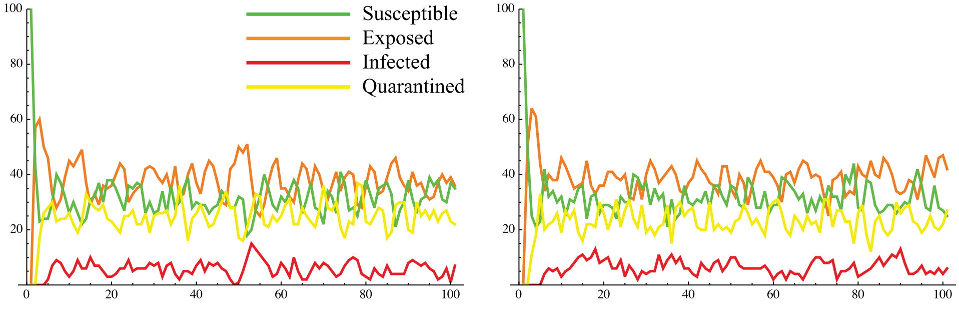
- Finally, the probability to receive an email from another computer will be defined as follows:

$$pV_{ij}^t = \begin{cases} \dfrac{1}{d_i}, & \text{if } v_j \text{ and } v_i \text{ are neighbor nodes,} \\ 0, & \text{if } v_j \text{ and } v_i \text{ are not neighbor nodes.} \end{cases}$$

Taking into account these values and considering a population of 100 hosts in which the percentage of users belonging to classes A, B, C and D are 5%, 30%, 45% and 20%, respectively, then the evolution of the different compartments is shown in Fig. 3(a) when a random computer network is considered. In Fig. 3(b) the evolution of the system is given when the computer network is defined by means of a complete graph (i.e. each pair of hosts is connected).

Note that the results obtained show a similar behavior in both situations; as a consequence it seems that the topology of the computer network does not affect to the final evolution of the compartments.

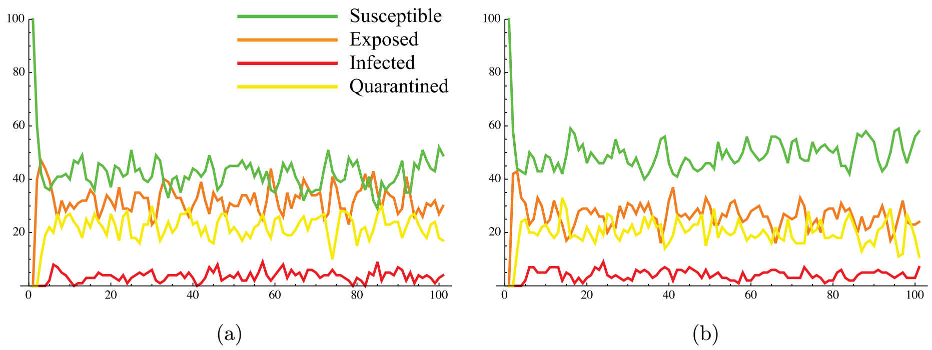
Moreover, the increase of the percentage of population awareness of security is reflected in the evolution of the compartments: The number of susceptible hosts increase and the number of quarantined and exposed hosts decrease. This situation is shown in Fig. 4 where the evolution is computed from different percentages of host classes and in Fig. 5, where homogeneous population is considered (that is, all users belong to the same class).



(a)

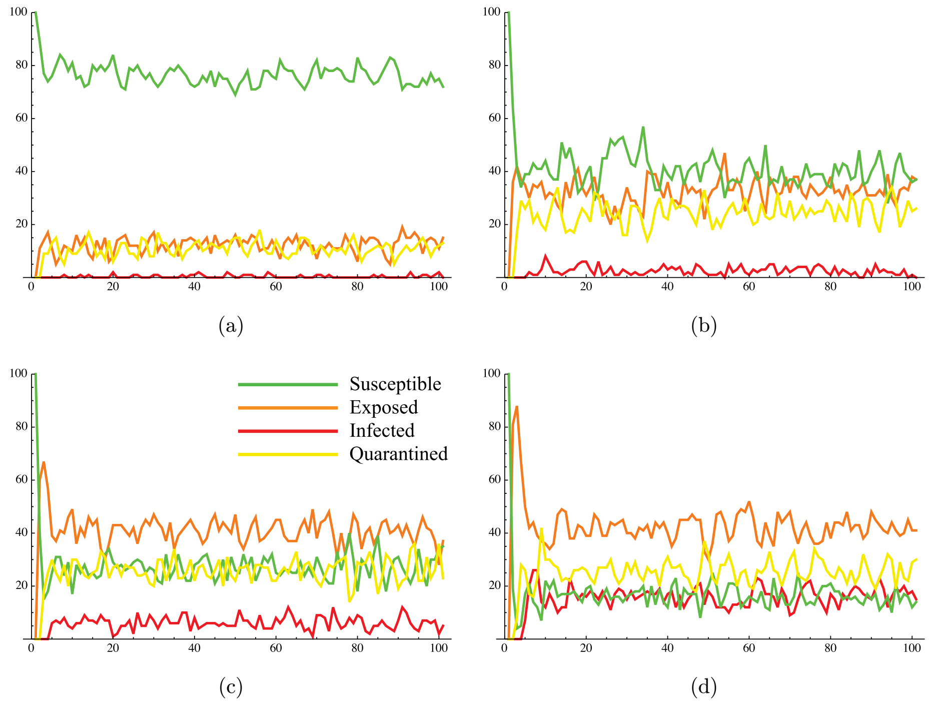
(b)

Fig. 3. (Color online) Evolution of the four compartments for $n = 100$ and during 100 iterations. (a) When a random network is considered; (b) When a complete network is taken.



(a) (b)

Fig. 4. (Color online) Evolution of the four compartments in a complete network for $n = 100$ and during 100 iterations. (a) A = 25%, B = 30%, C = 35% and D = 10%; (b) A = 40%, B = 30%, C = 20% and D = 10%.



(a) (b) (c) (d)

Fig. 5. (Color online) Evolution of the four compartments in a complete network for $n = 100$ and during 100 iterations. (a) A = 100%, B = 0%, C = 0% and D = 0%; (b) A = 0%, B = 100%, C = 0% and D = 0%; (c) A = 0%, B = 0%, C = 100% and D = 0%; (d) A = 0%, B = 0%, C = 0% and D = 100%.

If we reduce the length of the dormant period and latent period, the number of infectious hosts increase as is shown in Fig. 6. It is due to the fact that the probability to be detected increases with the life cycle of the computer virus; consequently, if we reduce such life cycle, the countermeasures do not act and the host becomes

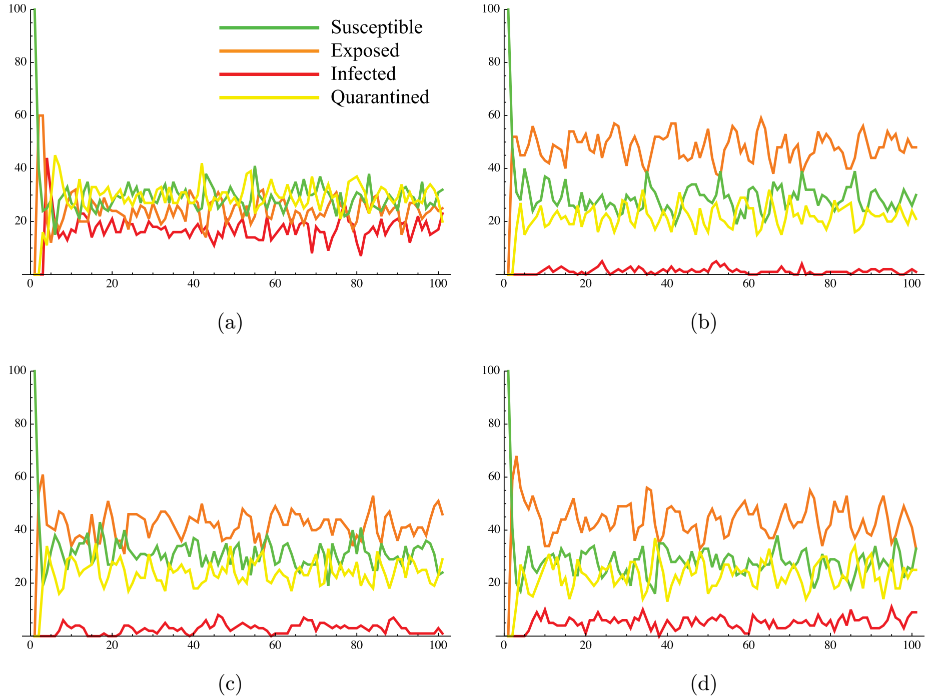


Fig. 6. (Color online) Evolution of the four compartments in a complete network for $n = 100$ and during 100 iterations. (a) $tD = tL = 1$; (b) $3 \leq tD,\ tL \leq 7$; (c) $1 \leq tD \leq 2,\ 3 \leq tL \leq 7$; (d) $3 \leq tD \leq 7$, $1 \leq tL \leq 2$.

infectious. Moreover, in Fig. 7 the evolution of both, the number of host safe from computer virus and the number of computers hosting the computer virus (in any of its stages) is presented. As is shown, the number of hosts without the computer virus is greater than others and this difference decreases as the length of the dormant and latency period increases.

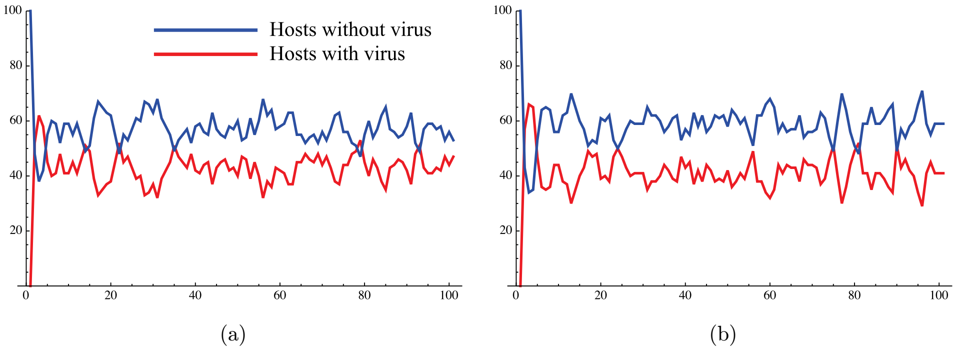


Fig. 7. (Color online) Evolution of the number of hosts with and without the computer virus. (a) $1 \leq tD \leq 2$, $2 \leq tL \leq 4$; (b) $tD = tL = 1$; (c) $3 \leq tD$, $tL \leq 7$; (d) $1 \leq tD \leq 2$, $3 \leq tL \leq 7$.

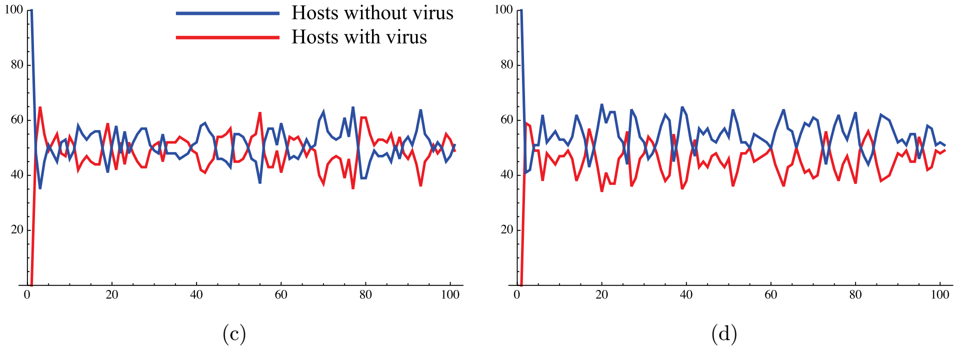


Fig. 7. (*Continued*)

6. Analysis of the System Equilibrium

For an epidemiological system, the disease-free equilibrium corresponds to a steady state of the population without disease (i.e. is the point at which no disease is presented in the population). Moreover, the endemic equilibrium is the steady state meaning that the disease will establish itself and remain in the population (i.e. is the point at which the disease persists in the population). As is well known, the epidemic models based on differential equations and difference equations (the great majority of the models proposed in the literature) are useful for estimating these two equilibria and their stability by means of the study of the eigenvalues of the Jacobian matrix. This approach is not possible for ultra-discrete models based on CA whose local transition functions are boolean functions depending on probabilistic parameters.

Let $\Upsilon(t)$ be the number of infected nodes at time t, that is, $\Upsilon(t) = E(t) + I(t)$ where $E(t)$ (respectively $I(t)$) are the total number of exposed (respectively infectious) computers at time t. Moreover, set $\Lambda(t) = S(t) + Q(t)$ the number of non-infected nodes (susceptible and quarantined) at time t. The equilibrium steady states are characterized as follows: There exists a step time t_0 such that:

$$\Upsilon(t+1) - \Upsilon(t) = 0, \tag{1}$$

$$\Omega(t+1) - \Omega(t) = 0, \tag{2}$$

for every $t \geq t_0$. In our case, Eq. (1) can be substituted by a condition stated that the probability that all nodes will be infected at a time $t \geq t_0$ will be zero. Suppose that the topology of the network is defined by means of a complete graph (this assumption looks like the situation considered in the majority of epidemic models based on differential equations), then a simple (but long) calculus shows that the last stated condition is:

$$\Gamma_i^t + (1 - \Gamma_i^t)\frac{1-(1-n)^n}{n(1-n)^n} = 0,$$

for every $1 \leq i \leq n$, where $\Gamma_i^t = \alpha_i^t + c_i^t\beta_i^t(1 - pF_i^t) - c_i^t\alpha_i^t\beta_i^t(1 - pF_i^t)$. As

$$\lim_{n\to\infty} \frac{1 - (1-n)^n}{n(1-n)^n} = 0,$$

when the number of nodes is very high (a realistic assumption) then the equilibrium points are defined by the following system of equations:

$$\alpha_i^t + c_i^t\beta_i^t(1 - pF_i^t) - c_i^t\alpha_i^t\beta_i^t(1 - pF_i^t) = 0, \quad 1 \leq i \leq n. \tag{3}$$

Consequently, the disease-free equilibrium is given by:

$$\alpha_i^t = \frac{c_i^t\beta_i^t(1 - pF_i^t)}{c_i^t\beta_i^t(1 - pF_i^t - 1)}.$$

If we suppose that $\alpha_i^t \approx 0$ (it is a reasonable assumption to make that the probability to acquire a computer virus from an external device can be considered negligible), then

$$c_i^t\beta_i^t(1 - pF_i^t) = 0,$$

that is, the disease-free equilibrium is reached when $\beta_i^t = 0$ or $pF_i^t = 0$ for every $1 \leq i \leq n$.

Due to the probabilistic nature of the system when $\beta_i^t \neq 0$ and $pF_i^t \neq 0$, the numerical analysis shows that the endemic equilibrium is not reached although there are fluctuations of the number of infected individuals constrained to an interval whose amplitude depends on β_i^t and pF_i^t.

Moreover, another consequence can be derived from the system (3): The propagation of a computer virus strongly depends on the probability to download infected files from web pages and the capacity of the security software installed in the computer to detect and block the virus.

7. Conclusions

In this work a model to simulate the propagation of computer virus through a computer network is introduced. It is based on the use of CA on graphs and four classes of host are considered: Susceptible, exposed, infectious and quarantined. It is a novel model in the sense that, to the best of our knowledge, there is no any other model proposed in the scientific literature, to simulate the spreading of computer virus (it does not occur with other type of malware such as computer worms). In this model several parameters are considered related to the cycle life of the computer virus (stages, periods, etc.), to the countermeasures implemented in the hosts, and to the behavior of the users. The results obtained seem to be in agreement with the reasonable behavior. Furthermore, the basic hints about the conditions to avoid infection propagation are the following: (1) Avoid the download of files from suspicious web pages; and (2) Install and updated an efficient antivirus software.

Acknowledgments

This work has been supported by Ministry of de Economy and Competitiveness (Spain) and European FEDER Fund under projects ACAC (MTM2008-02773) and TUERI (TIN2011-25452).

References

1. Y. Xiao, F. H. Li and H. Chen, *Handbook of Security and Networks* (World Scientific Publishing, Singapore, 2011).
2. J. Bradley, S. Gilmore and J. Hillston, *J. Comput. Syst. Sci.* **74**, 1013 (2008).
3. S. Kondakci, *Simulat. Modelling Pract. Theory* **16**, 571 (2008).
4. B. K. Mishra and D. Saini, *Appl. Math. Comput.* **187**, 929 (2007).
5. B. K. Mishra and D. Saini, *Appl. Math. Comput.* **188**, 1476 (2007).
6. J. Piqueira, A. Vasconcelos, C. Gabriel and V. Araujo, *Comput. Secur.* **27**, 355 (2008).
7. H. Yuan and G. Chen, *Appl. Math. Comput.* **206**, 357 (2008).
8. A. L. Bauer, C. A. A. Beauchemin and A. S. Perelson, *Inform. Sci.* **179**, 1379 (2009).
9. A. Martín del Rey, *A Computer Virus Spread Model Based on Cellular Automata on Graphs — Proc. of IWANN 2009, Part II*, eds. S. Omatu *et al.*, Lecture Notes in Computer Science, Vol. 187 (Springer, 2007), p. 929.
10. Y. Song, G.-P. Jiang and Y. Gu, Modeling malware propagation in complex networks based on cellular automata, in *Proc. IEEE Asia Pacific Conference on Circuits and Systems*, 2008, p. 259.
11. T. Toffoli and N. Margolus, *Cellular Automata Machines: A New Environment for Modeling* (MIT Press, Cambridge, MA, 1987).
12. W. Wolfram, *A New Kind of Science* (Wolfram Media, Champaign, IL, 2002).
13. S. Eubank, V. S. A. Kumar and M. Marathe, *Epidemiology and Wireless Communication: Tight Analogy or Loose Metaphor? — Proc. Bio-Inspired Computing and Communications*, eds. K. T. D. Eames and J. M. Read, Lecture Notes in Computer Science, Vol. 5151 (Springer, 2008), p. 91.
14. S. Goel and J. S. Gangolly, *Int. J. Inf. Manage.* **27**, 266 (2007).
15. J. Li and P. Knickerbocker, *Comput. Secur.* **26**, 338 (2007).
16. M. Rice, J. Butts, R. Miller and S. Shenoi, *Int. J. Critical Infrastructure Protection* **3**, 118 (2010).
17. J. Aycock, *Computer Viruses and Malware*, Advances in Information Security, Vol. 22 (Springer Science + Business Media, 2006).