Infection Tracing to limit the spread of Computer Virus

Matteo Antonio Inajetovic

ID: 0000949648
E-mail: matteo.inajetovic@studio.unibo.it

Abstract. In this project work, the spread of computer virus is modelled with a SIRS model and studied in the context of a computer network generated with the Preferential Attachment method. An Infection Tracing Tool is proposed to limit the virus diffusion in critical conditions and evaluated exploiting network analisys measures.

1 Introduction

Nowadays, computer virus is one of the most dangerous weapons in internet. Considering the strong development of information technology and its application in our daily life, in almost all research and industrial fields, is necessary to develop and realize methods to detect virus infections and limit their diffusion in computer networks. Indeed, even if antivirus software is the main defence against viruses, the antivirus technology is not able to predict their evolution trend and cannot provide strategies to prevent them. The strong will to understand and avoid the spread mechanism of computer viruses has encouraged the implementation of many epidemic models.

One of the most important contribution to this research field was done by Han and Tan (2010), who presented a mathematical model about the dynamical diffusion of viruses in the internet and more thoroughly, a precise study about global and local stability of the proposed system [1]. Gan et al. (2012), focusing on the propagation of computer virus under human intervention, showed how prevention is more important than cure, and that a higher disconnecting rate from the Internet affects significantly the virus diffusion [2]. Deng and Liu (2012), with the realization of an epidemic model of computer virus on scale-free network with nonlinear infectivity, described how the outbreak of the virus is entirely determined by the critical threshold [3].

The aim of this project work is to consider an epidemiological model for a dynamical virus diffusion and to evaluate the efficiency of a tracing technique to inhibit the epidemics.

2 Preliminaries

To better understand the work project, some important definitions and theoretical aspects are going to be introduced.

2.1 SIRS model

The SIRS epidemiological model is an extension of the basic SIR model, which is a simplified representation of how an infection spreads across a population over time. It can be applied to many epidemiological fields, such as health, sociology, marketing and informatics [4]. In this model the population, generally constant, is divided in three, mutually exclusive, compartments: susceptible (S), infected (I) and resistant (R). Given the compartments, is important to describe how their size changes over time:

$$\frac{dS}{dt} = -\beta SI \tag{1}$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

where the infection rate is β and the recovery rate is γ. The crucial factor of this model is the reproduction number R_0, defined as the average number of secondary infections occurring when one infective is introduced into a completely susceptible population.

When $R_0 < 1$ the virus spread affects only a portion of the population and the infection dies out. The time needed to extinguish the infection depends on the R_0's size. Otherwise, if $R_0 > 1$, the infection will spread to the entire population.
The SIRS model instead, takes into account the probability δ of resistant computers to become susceptible again:

$$\frac{dS}{dt} = -\beta SI + \delta R \tag{2}$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$
$$\frac{dR}{dt} = \gamma I - \delta R$$

In both models, the demography can be considered by adding to the equations the birth rate, the death rate and the death rate due to the virus [1, 4]. Obviously, adding new features to the model complicates the computation of the R_0 parameter.

2.2 Preferential Attachment – The Barabàsi-Albert model

In real networks, such as computer networks, new nodes usually link themselves to the more connected nodes, rather than randomly choose their neighbors like it happens in random networks. This lead Barabàsi and Albert to apply the Preferential Attachment method to the growth of networks, capturing the real network's scale-free power-law distribution [5]. In detail, the Barabàsi-Albert method starts with a network made up of m_0 initial nodes and links chosen arbitrarily (each node has at least one link). Then it follows these steps:

- (Growth) At each timestep a new node is added with m ($\leq m_0$) links that connect the new node to m nodes already in the network.
- (Preferential attachment) The probability Π(k) that a link of the new node connects to node i depends on the degree k_i as:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

Therefore, this probabilistic mechanism generates, after t timesteps, a network with $N = t + m_0$ nodes and $mt + m_0$ links. In summary, this method catches all the main features of a computer network, such as the "rich get richer" phenomenon, the scale-free networks' formation and their degree distribution.

2.3 Infection Tracing

In the epidemiologic field, an early detection of infection cases, as well as the source of infection, is necessary to identify individuals at risk effectively. By using the Infection Tracing technique, each infected agent is linked to those which caused its infection, and additionally, to those it will infect, building the, so called, Infection Graph or Transmission Network. Defining links is not difficult since each directed edge of the graph reflects a transmission event and, moreover, the resulting graph will be tree-like and won't contain any kind of loops.

3 Network and Model Description

This project work relies on the base model "Virus on a Network" [6, 7], which shows the spread of a virus through a network. Each node of the network represents a computer and may be in one of three states: susceptible, infected, or resistant. Given the assumption in which the virus is spreading by emailing itself out to everyone in the computer's address book, the directed link (i,j) represents the chance, for a computer i, to send an email to the computer j. Each time step, the susceptible computers have a probability (linked to the `virus-spread-chance` parameter) to be infected by their infected neighbours which can send them an infected email. In this model, is not possible to detect immediately an infection, instead, there's a periodic check determined by the `virus-check-frequency` parameter. Once an infected computer has been detected it also has a probability to be recovered (`recovery-chance`). For each simulation, the network is generated with the Preferential Attachment method (Fig. 1), because real computer networks usually exhibit a "scale-free" link-degree distribution, rather than the one used in the base model. Furthermore, for each resistant computer, the probability to lose its resistance and becoming susceptible again was also included (`loss-immune-rate`), transforming the SIR model into a SIRS one. This phenomenon might correspond to a missed renewal of the antivirus software. The study of the tipping point is more complex than the usual SIRS model, since the `virus-check-frequency` parameter affects the computation of the R0 factor: if the former is too low the virus is always stopped, even if the latter is bigger than 1; otherwise, if it is too high the system will always be in the epidemic state, with a periodic solution. So, this study is focused on the critical situation where the recovery rate is lower than the infection one and where the `virus-frequency-check` is not low enough to avoid the periodic behaviour of the SIR curves. Given this critical condition, the aim of this work is to implement a tracing technique to stop the virus spread.

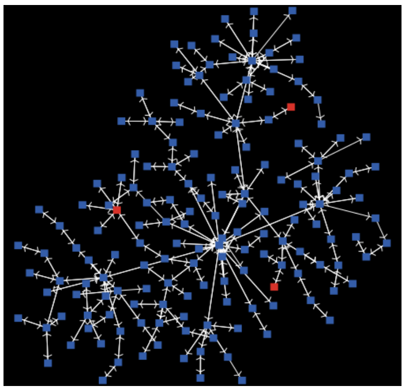

Fig. 1. Network generated with Preferential Attachment (before the simulation starts).

4 Infection Tracing Method

The relevant part of the project work consists in the realization of an Infection Tracing tool, a method able to stop the virus spread and reduce its damages over the computer network. To study its effectiveness, it was applied to different scenarios, mainly grouped by two macro-cases. In the first one the network has a fixed population N without "deaths" and "births" of nodes, such that it can be described by a basic SIRS model. Instead, in the second case, the number of computers changes during the simulation, since demographic features, such as the lethality of the virus, the connection of new computers to the Network and the death rate of the agents, are considered. Looking at the first scenario, if the tipping point is passed, the epidemics could never stop and the SIR curves will assume a periodic evolution. With the application of the Infection Tracing Tool this kind of situation can be avoided and thus, the virus spread is stopped.

In the second scenario, according to the assumption where computers that die lose their ability to contact their neighbors, the virus will always die out. Nevertheless, the realized tool is able to limit the number of deaths caused by the virus and its diffusion over the whole network.

Before getting into the technical part of the experiments, is important to specify some assumptions: (a) all infected computers have equal resistance against the virus (the parameter `virus-death-rate` is the same for all the individuals), (b) even restored computers can become susceptible again with a certain rate and (c) the system's condition is always the critical one, described in the section 3.

4.1 Infection Tracing Tool's description

The tracing technique used to gather knowledge about the network and its evolution during the simulation is, as already seen, the infection tracing one. This method can trace all sources of infection for each individual that have been infected and the resulting tracing-graph will be a directed one, in order to give more insights about the centrality of nodes and their role in the diffusion.

In this project work's environment, the infection tracing can be applied because for each computer is possible to retrieve the e-mail that generated the infection and thus, the sender. The realized method acts in two main phases: the detection and the isolation one. For each simulation, the tool will label a percentage of the initial number of computers (linked to the `tool-efficiency` parameter) with the tag `control?`. Then for all the tagged individuals it computes their centrality value and store them in a sorted list, from which is possible to obtain a centrality threshold to distinguish relevant nodes from the irrelevant ones. The tagged computers cannot loose their resistance since they're being monitored. Once that the simulation starts, at each virus check the Infection Trancing Tool stores all the sources of infection and all the infected nodes to build the Infection Graph: if one of the tagged individuals with a significant centrality score has been infected, then the tool will isolate all the tagged individuals, avoiding them to send other emails to their contacts and consequentially, to spread the virus. At the end of the simulation, the infection graph can be visualized and studied to evaluate the decisions made and verify the hypothesis.

Observation. It is clear how the `tool-efficiency` parameter will influence the system's evolution: if it's really low it will be useless, if it's too high, the developed method would be linked to a non-realistic scenario. For those reasons is important to tune it in a reasonable way. Now let's look at the different experiments.

4.2 Static Population Scenario

In order to study this variant of the model, where the virus is not lethal, all the parameters linked to the "death" or the "birth" of an individual are to be considered and the number of computers N is fixed during all the simulation. If the system is in the critical condition (assumption (c)), the virus can never be stopped and brutally affects the network. The effectiveness of the Infection Tracing Tool can be observed by running two simulations, one without any kind of action to limit the diffusion and another where the tool is active (Fig. 2).

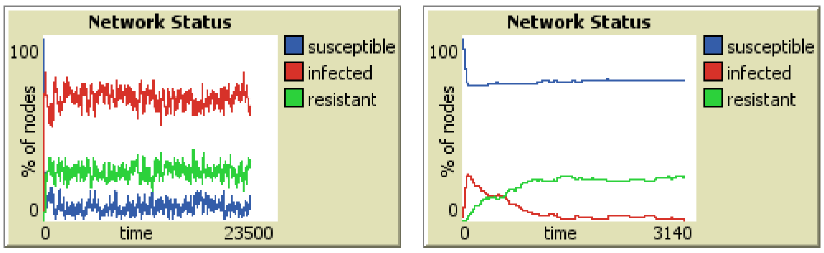

Fig. 2. Network's evolution in both cases: with (on the right) and without (on the left) the Infection Tracing Tool. (`tool-efficiency` = 80%)

Therefore, exploiting the tool, the virus can be stopped even if the network status passes the tipping point. Indeed, the number of infections is reduced significantly (about one order of magnitude, from thousands to less than one hundred) just like the number of nodes involved in the diffusion (from almost the whole network infected to less than a half).

In order to understand whether the position of the initial outbreak nodes influences the evolution of the system, it's possible to choose, before the simulation starts, hub nodes (with a higher node degree) or nodes at the periphery of the network. This option doesn't modify significantly the results.

If two neighbours are not taken into account by the tool's tagging, they can, potentially, continue to spread the virus periodically one to the other because they're not monitored. So, even if this periodic spread could last infinitely, the tool is still efficient since it will limit the virus diffusion in a certain region without allowing it to affect the rest of the network. In this kind of rare situations, the `tool-efficiency` parameter is crucial: reducing the sample of monitored computers, the probability to have this kind of "blocked" regions of infected nodes increases. Obviously, if the sample is almost equal to the number of nodes, this phenomenon never happens.

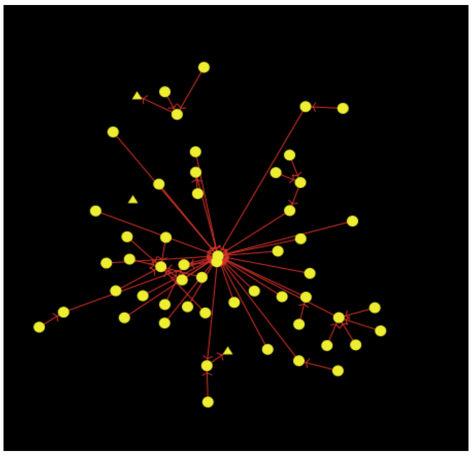

Fig. 3a. Infection Graph for the network evolution using the tool. The triangles represent, for a visualization purpose, the initial outbreak nodes. In this case it has 56 nodes involved in the virus diffusion (over 160 computers) and 62 single infections. Without the tool, it would have all the nodes involved and thousands of infections.

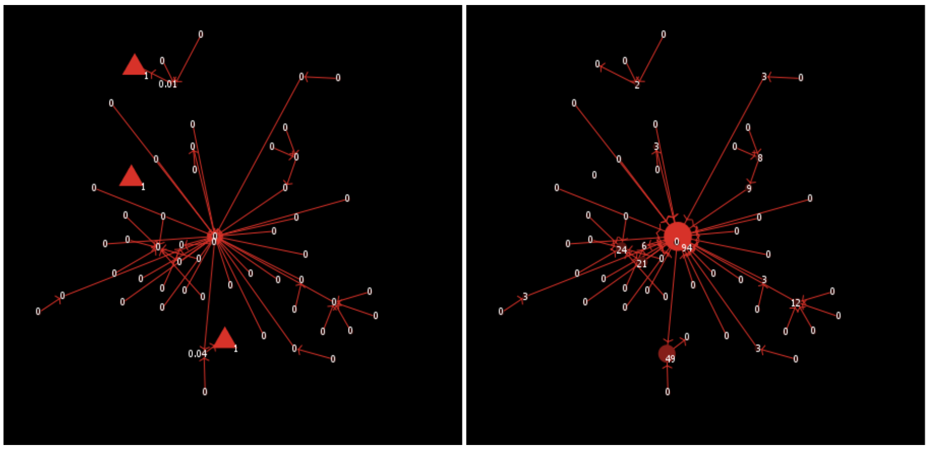

Fig. 3b. Visualization of the Infection Graph by means of Eigenvector Centrality (on the left) and Betweenness Centrality (on the right).

The Infection Graph (Fig. 3a) shows the diffusion of the virus over the network and returns both the number of nodes involved in the infection and the number of single infections. Furthermore, with the Eigenvector Centrality [8], the tool is able to detect the initial outbreak nodes and also the most relevant nodes in the diffusion, thanks to the Betweenness Centrality [9] (Fig. 3b).

4.3 Dynamic Population Scenario

In the second scenario, as already mentioned, the virus can “kill” the infected individuals according to the `virus-death-rate` parameter. Going deeply, this happens if infected computers have not been recovered before a certain amount of time, tuned by the `sick-tick-to-die` parameter. Furthermore, new computers can join the network during the simulation, with a chance equal to the `birth-rate` and some of them can also “die” (by means of leaving the network for some reason) depending on the `death-rate`. By adding these new features, the model changes its behaviour: the virus will stop at a certain time killing a huge part of the initial population.

By using the Infection Tracing Tool, the virus deaths can be reduced significantly, just like the number of infections (Fig. 4a and Fig. 4b). Another difference, with respect to the static population scenario, consists in a strong change of the Degree Distribution (Fig. 4c), due to computers leaving or joining the network.

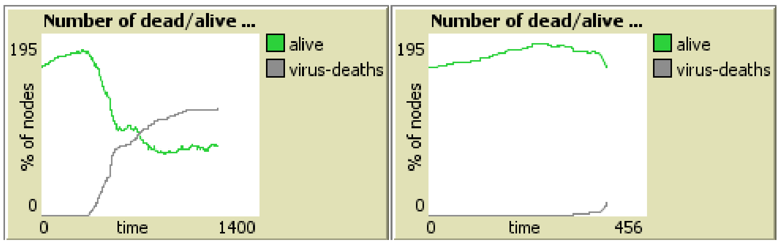

Fig. 4a. Plots concerning the evolution of the number of “alive” computers and the number of deaths caused by the virus. Looking at the plot on the right, it can be seen how the tool stops the infection faster than the basic model with a very little number of virus deaths. Parameters used: tool-efficiency = 80%, sick-ticks-to-die = 300, virus-death-rate = 30%, birth-rate = 8%, death-rate = 0.3%.

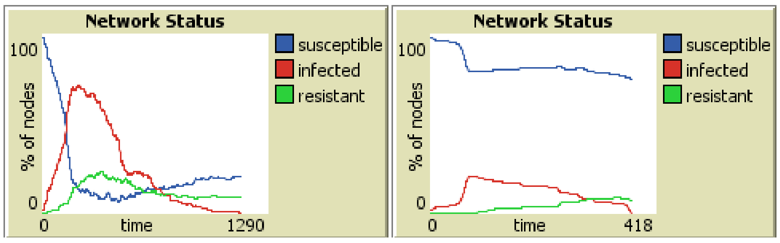

Fig.4b. Network’s evolution in both cases: with (on the right) and without (on the left) the Infection Tracing Tool.
Parameters used: tool-efficiency = 80%, sick-ticks-to-die = 300, virus-death-rate = 30%, birth-rate = 8%, death-rate = 0.3%.

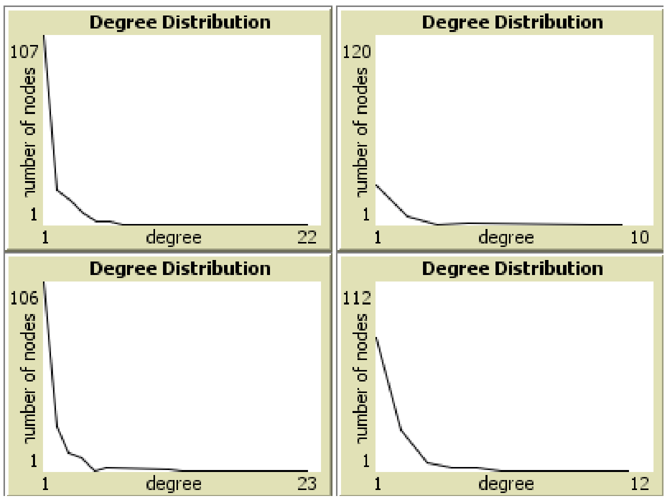

Fig. 4c. Looking at the two figures on the top, describing the simulation of the network without the tool, the virus affects the network structure (the most part of the population will have degree equal to zero). Instead, in the simulation linked to the two figures at the bottom, the tool limits the virus effect on the network's degree distribution.

Just like the static population scenario, it is possible to visualize the Infection Graph and compute the number of infections, the number computers involved in the diffusion captured by the tool and the centrality measures (Fig. 5a and Fig. 5b).

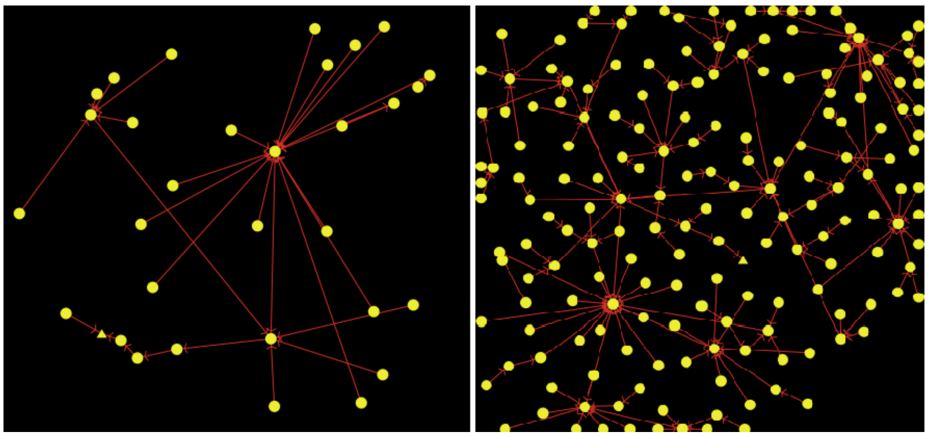

Fig. 5a. Infection Graph (left) with the tool: number of infections = 31, number of computers involved = 32.
Infection Graph (right) without the tool (computed only for a comparison purpose): number of infections = 196, number of computers involved = 180.
(initial-outbreak-size = 1).

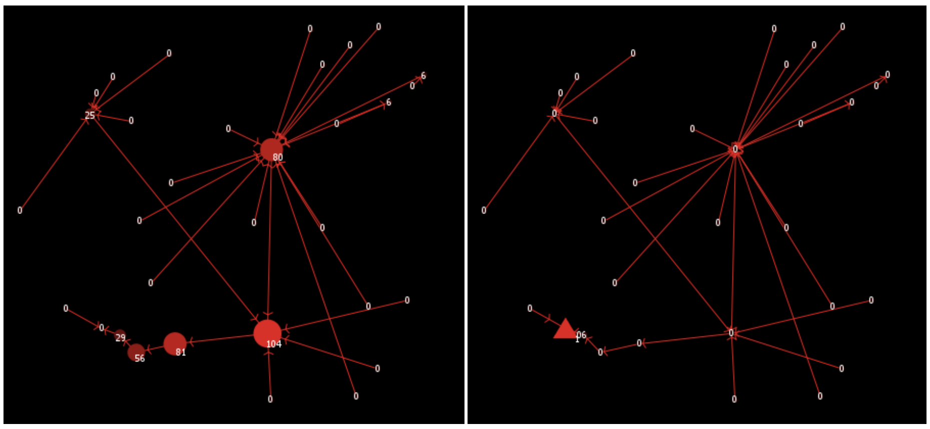

Fig. 5b. Betweenness and Eigenvector Centrality plots for the Infection Graph of Fig.5a.

Finally, it is possible to state that the application of the Infection Tracing Tool leads to a significant reduction of the virus' effects on the Network, as can be seen looking at other four simulations' results from Table 1.

Table 1. initial-outbreak-size = 3, number-of-nodes = 160, virus-spread-chance = 5%, virus-check-frequency = 5 ticks, recovery-chance = 1%, tool-efficiency = 80%, age-to-die = 1000 ticks, sick-ticks-to-die = 300 ticks, birth-rate = 10%, virus-death-rate = 30%, death-rate=0.3%.

Simulation	Population	Tool's usage	Infected nodes	Number of infections	Virus Deaths
1)	Static	No	160	(*)	X
2)	Static	Yes	37	34	X
3)	Dynamic	No	203	224	115
4)	Dynamic	Yes	47	44	25

(*) The number of infections for the Static scenario grows almost linearly w.r.t time and the simulation can never stop, as explained in the previous sections.

5 Conclusion

In conclusion, even if the realized model is quite abstract and simplified by many assumptions, the application of the developed Infection Tracing Tool turns out to be effective in both macro-scenarios addressed, in normal and stressed situations (like the one where the infectious rate is higher than the recovery one). This work could be improved by studying more complex models and facing more realistic and specific environments.

References

1. X. Han and Q. Tan, Dynamical behavior of computer virus on Internet. Applied Mathematics and Computation, 217: 2520–2526, 2010
2. C. Gan, X. Yang, W. Liu, Q. Zhu and X. Zhang, Propagation of computer virus under human intervention: a dynamical model. Discrete Dynamics in Nature and Society, Hindawi Publishing Corporation ID 106950, 2012.
3. C. Deng and Q. Liu, A computer virus spreading model with nonlinear infectivity on scale-free network, in International Conference on Information Sciences, Machinery, Materials and Energy. Atlantis Press,1684–1688, 2012.
4. Rodrigues, Helena Sofia (2016). Application of SIR epidemiological model: new trends, International Journal of Applied Mathematics and Informatics, 10: 92—97
5. Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. science, 286(5439):509–512, 1999.
6. Stonedahl, F. and Wilensky, U. (2008). NetLogo Virus on a Network model. http://ccl.northwestern.edu/netlogo/models/VirusonaNetwork Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
7. Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
8. Zaki, Mohammed J.; Meira, Jr., Wagner (2014). Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press. ISBN 9780521766333
9. Brandes, Ulrik (2001). "A faster algorithm for betweenness centrality". Journal of Mathematical Sociology. 25 (2): 163–177.