Modeling Malware Spreading Dynamics

Michele Garetto
Dipartimento di Elettronica
Politecnico di Torino
Torino, Italy 10129
Email: garetto@polito.it

Weibo Gong
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003
Email: gong@ecs.umass.edu

Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
Email: towsley@cs.umass.edu

***Abstract*— In this paper we present analytical techniques that can be used to better understand the behavior of malware, a generic term that refers to all kinds of malicious software programs propagating on the Internet, such as e-mail viruses and worms. We develop a modeling methodology based on *Interactive Markov Chains* that is able to capture many aspects of the problem, especially the impact of the underlying topology on the spreading characteristics of malware. We propose numerical methods to obtain useful bounds and approximations in the case of very large systems, validating our results through simulation. An analytic methodology represents a fundamentally important step in the development of effective countermeasures for future malware activity. Furthermore, we believe our approach can help to understand a wide range of "dynamic interactions on networks", such as routing protocols and peer-to-peer applications.**

I. Introduction

The easy access and wide usage of the Internet makes it a prime target for malicious activity. In particular, the Internet has become a powerful mechanism for propagating malicious software programs designed to annoy (e.g., deface web pages), spread misinformation (e.g., false news reports or stock quotes), deny service (e.g., corrupt hard disks), steal financial information (e.g. credit card numbers), enable remote login (e.g., Trojan horses), etc. The two most popular ways to spread such malicious software are commonly referred to as worms (like the Code Red) and email viruses (like the infamous Melissa and Love Bug). However it is increasingly difficult to distinguish malicious software programs using these terms. For example, the recent Nimda attack was especially vicious because it combined both attack methods. For this and other reasons we will refer to all malicious programs propagating on the Internet as *malware*.

Although malware has resulted in economic losses, so far they have been mostly nuisances. However it is expected that future malware will be more virulent and, thus, result in significantly greater damage. A recent document from CERT [12] reports on increasing attempts to compromise routers along with end-hosts as well as other dangerous trends.

Currently, malware are reverse engineered at some computer security organizations. Analysis of the malware signature is then broadcast to system administrators for countermeasure deployment. However, for the most part, it is not possible to control the spreading of unknown malware, that can quickly propagate through the network, infecting many machines before the severity of the situation is recognized. To date there appears to be no well defined methodology for predicting the behavior of malware. For example, one would like to be able to estimate whether or not the malware is sufficiently potent to infect the entire Internet in the absence of countermeasures. If the answer is positive, one would like to determine the required effectiveness of countermeasures in order to control the spread. Finally, one would like to compare different network architectures with respect to their vulnerability to malware infection, in order to prevent major catastrophic events.

The goal of our work is to develop mathematically-based methodologies that can be used to better understand the behavior of malware, including their spreading characteristics. To this purpose we build a stochastic model based on *Interactive Markov Chains* (IMC) that provides a probabilistic analysis of the system. Although we have focused on the propagation of e-mail viruses, the approach is general enough to be adapted to describe other kinds of malware. While the exact solution is computationally too expensive in the case of a large system, the exact details of the distributions are not crucial, so that rough estimates may suffice for prediction purposes. Thus we develop algorithms to predict gross-level system-wide behavior and obtain useful bounds and satisfactory approximations.

Very little work has appeared so far in the literature on modeling computer viruses. An investigation was carried out in the early 1990s at IBM Research via both simulation and analysis by means of standard epidemic models [4]. In this work it was pointed out the difficulty in extending approaches suitable to analyze fully connected graphs to arbitrary topologies, where the propagation of a virus can exhibit characteristics not easily described by simple system-wide equations. The study presented in [6] was conducted using simulation experiments that show the impact of different strategies of immunization on certain types of networks. Outside of networks, the analogous problem of spreading of a disease within a population has been the object of mathematical epidemiology for over a century. A book of lecture notes that covers the main stochastic techniques used in the area is [5]. The more general problem of "dynamic interactions on networks" has been studied in a number of fields (physics, biology, economics, sociology) using a variety of techniques, so that it is not possible here to provide a comprehensive survey of previous approaches. Interactive Markov Chains have been suggested for modeling vulnerabilities in power systems [7] and communication networks [8]. In this paper we adopt a technique called "influence model" originally introduced in [9]. Particularly important to our work is also a methodology drawn from statistical physics

that has been recently applied to percolation and epidemics on networks [13], [14].

The rest of the paper is organized as follows: we describe our modeling approach in Section II. In Section III we show how the problem of estimating the final size of a malware infection maps onto a percolation problem, and we present an algorithm to solve the percolation problem on the small-world graph. In Section IV we propose our solution to derive analytically the state evolution of the system. We suggest directions to extend our work in Section V. Finally, we conclude the paper in Section VI.

II. Modeling approach and assumptions

We developed a stochastic model of malware propagation based on the *Interactive Markov Chains (IMC)* framework. An IMC consists of a network of nodes specified by a directed graph $G = (V, E)$. A node on the graph is also called a *site*. Each site has a *status* that evolves over time. We use instead the term *state* to refer to the collection of statuses of all of the sites at a given time. The status at a site evolves according to an internal Markov chain, but with transition probabilities that depend not only on the current status of that site, but also on the statuses of the neighboring nodes. The overall system evolves according to a global Markov Chain whose state space dimension is the product of the number of statuses describing each site. Because of the exponential growth in the number of states, large IMCs are extremely difficult to solve numerically, even for a few tens of nodes, so that it is necessary to resort to discrete event simulations[1].

A special case of IMC called "influence model" has been recently proposed in [9] that provides a particular but tractable representation of dynamic interactions on networks. In the "influence model" it is possible to obtain the marginal status probabilities of each site by means of a transition matrix whose dimension is equal only to the sum of the dimensions of the local chains. Our model is based on the influence model technique, but we allow state transitions to occur in a more complicated way than what is described in [9].

The rest of the Section is organized as follows. In Section II-A we provide a brief background on the influence model following [9]. In Section II-B we describe how we adapted the influence model approach to the problem of virus propagation, including a discussion of the assumptions that we made.

A. The influence model

An influence model is defined as a discrete-time Markov process. For our application, we can assume that the behavior of the whole system is ergodic. Let $\boldsymbol{\pi_j}\ [k]$ be the status probability row vector of site j at a given time step k. If the site were isolated (i.e. not connected to the graph) it would follow a standard Markov chain, so that we could write $\boldsymbol{\pi_j}\ [k+1] = \boldsymbol{\pi_j}\ [k]\,\mathbf{P}$, where $\mathbf{P}$ is the transition matrix. The influence model allows an arbitrarily connected structure of sites defined by a weighted directed graph $G = (V, E)$, in which $w_{i,j}$ is the weight associated to the edge directed from i to j ($w_{i,j}$ is equal to zero if no edge exists from i to j). Each weight $w_{i,j}$ takes a value in the interval $[0, 1]$, and represents the amount of influence that i exerts on j relative to the total amount of influence that j receives, which is normalized to one: $\sum_{i=1}^{N} w_{i,j} = 1$ (let $N = |V|$ be the number of sites in the graph). In the influence model the evolution of each site is constrained to take the multi-linear form $\boldsymbol{\pi_j}[k+1] = \sum_{i=1}^{N} w_{i,j}\ \boldsymbol{\pi_i}[k]\,\mathbf{P}_{i,j}$ which can be interpreted as follows: at each time step, site j selects with probability $w_{i,j}$ one of the neighboring sites in the network (or it selects itself) to be its determining site for the next step. The transition matrix $\mathbf{P}_{i,j}$ (which has a number of rows equal to the number of statuses in i and a number of columns equal to the number of statuses in j) completely specifies the way in which site i influences site j. If we stack the status probability vectors $\boldsymbol{\pi_j}$ into a single row vector $\boldsymbol{\Pi} = [\ \boldsymbol{\pi_1}\ \boldsymbol{\pi_2}\ \ldots\ \boldsymbol{\pi_N}\ ]$ it is possible to write more compactly $\boldsymbol{\Pi}[k+1] = \boldsymbol{\Pi}[k]\,\mathbf{H}$, where $\mathbf{H} = \mathbf{W} \otimes \mathbf{P}_{i,j}$ is called the *influence matrix* and can be expressed as the (generalized) Kronecker product of $\mathbf{W} = \{w_{i,j}\}$, which is called the *network matrix*, and the matrices $\mathbf{P}_{i,j}$:

$$\mathbf{H} \triangleq \begin{bmatrix} w_{1,1}\,\mathbf{P}_{1,1} & \cdots & w_{1,N}\,\mathbf{P}_{1,N} \\ \vdots & & \vdots \\ w_{n,1}\,\mathbf{P}_{N,1} & \cdots & w_{N,N}\,\mathbf{P}_{N,N} \end{bmatrix} \tag{1}$$

Doing so we separate out the impact of network topology ($\mathbf{W}$) from the effect of local interactions ($\mathbf{P}_{i,j}$). We can recursively obtain the marginal status probabilities of all sites at any given time from the simple equation

$$\boldsymbol{\Pi}[k] = \boldsymbol{\Pi}[0]\,\mathbf{H}^k \tag{2}$$

where $\boldsymbol{\Pi}[0]$ is the initial sites configuration. The multi-linear form of the influence model leads to a highly tractable model with rich mathematical properties, as reported in [9].

B. Virus propagation model

Our stochastic model of malware propagation is based on the influence model paradigm and focuses on email viruses. We believe a similar approach can be adopted to study other forms of malware with different spreading characteristics.

Most email viruses work as follows. An email message containing the virus program as an attachment is sent. A certain amount of time elapses before the recipient reads the email message. At this time, he/she has to decide what to do with the content of the message. Opening the attachment executes the email virus program, which will use the recipient's address book and/or inbox to spread copies of itself to other email addresses, in addition to performing other malicious activities on the infected machine.

In order to model the dynamics underlying email virus propagation, we build a directed graph in which each node corresponds to an email address and the edges represent social or business relationships between addresses. Recent research suggests that social and business networks form a so-called small world graph [11]. Since email addresses represent a subset of the human population, it is reasonable to assume that the graph defined by email address books also forms a

[1] A network of 20 nodes, each with a binary status, already leads to a system with over a million states

small world graph. In particular we expect the resulting graph to exhibit two fundamental properties: a small characteristic path length and a high clustering coefficient (see [10]).

The influence model requires the assignment of a "weight" to each edge, such that the sum of all incoming edges into a node is one. This can be interpreted as follows: the weight $w_{i,j}$ represents the probability that during a time step a user checks if any message has been delivered from source address i to destination address j. This means that we assume the time interval between two successive emails is a geometrically distributed random variable independent for each source-destination address pair. A proper time step granularity must be chosen so that the sum of the weights of all incoming edges into a node is smaller than or equal to one. If smaller, a self-loop can be added to the node in order to reach the value of one. This trick can be used to model the behavior of users who have a different number of email contacts, in case we assign an equal probability to each edge.

Note that the weight associated with an edge is just the probability of reading the message, not yet deciding what to do with its content. If the email contains a virus, we need to distinguish the cases in which the user opens the attachment or not. For this purpose we introduce a 'click' probability c_i for each node on the graph (that may be different from node to node), and we assume that the decision whether to open or not the attachment occurs only once. In other words, if a user receives an email message containing the virus, with probability $(1 - c_i)$ he/she decides once and for all not to open it, and this decision will not occur again in the future in case new copies of the virus are received.

Our discussion so far implies the necessity of introducing at least three statuses at each site, that we call *susceptible* (S), *infected* (I) or *immune* (M). *Susceptible* means that the site can be potentially infected by the virus, but no messages containing the virus program have been yet delivered to it, or the user has not yet checked for new e-mails; *Infected* means that the user has opened the attachment and the virus has successfully infected the machine sending copies of itself to all of the neighboring sites. We assume that a site, once infected, remains in this status forever; *Immune* means that the site cannot be infected by the virus. This can be due to several reasons: i) the virus program cannot execute on the hosting machine; ii) the site received a copy of the virus, but the user decided once and for all not to open the attachment; iii) the machine was initially vulnerable, but was later 'immunized' thanks to the effect of countermeasures, such as an anti-virus upgrade, a patch, or simply because the user has been warned not to open the attachment. iv) the site will never be reached by the virus (see Section III for an explanation of why this can occur); Once immunized, a site remains in this status forever.

The status transitions allowed within a site are shown in the oval of Figure 1, that shows an example of a network graph with five sites, expanding the internal structure of the rightmost site.

Unfortunately, it is not possible to formulate this model in terms of the influence model described in Section II-A. In particular, it is not possible to preserve the multi-linear form that leads to equation (2). In the influence model, the next

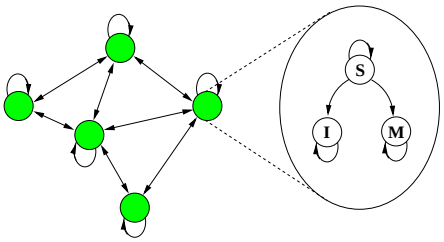

Fig. 1. Graphical representation of the model

status of a site that gets influenced by a neighbor is determined only by the current status of that neighbor, without depending concurrently on its present status. Our application requires a *status-dependent* influence model, that differs substantially from the influence model of [9], which is inherently *status-independent*. In our model, sites are influenced by neighbors only if they are still susceptible. Sites that are infected or immune are not influenced by other sites and do not change their status. As a consequence, changes in the state of the system are only due to residual contacts between still susceptible sites and already infected sites: when there are no more of these contacts anywhere on the graph, the system stops evolving.

Our stochastic model can be formulated as follows. For each site j, let $P_{S_j}(k)$, $P_{I_j}(k)$, $P_{M_j}(k)$ be the probabilities that, at time k, site j is susceptible, infected or immune, respectively. The status evolution of site j is described by the following system of recursive equations:

$$\begin{cases} P_{I_j}[k+1] &= P_{I_j}[k] + \sum_{i=1}^{n} w_{i,j}\, c_j\, P_{I_i S_j}[k] \\ P_{M_j}[k+1] &= P_{M_j}[k] + \sum_{i=1}^{n} w_{i,j}\, (1-c_j)\, P_{I_i S_j}[k] \\ P_{S_j}[k+1] &= 1 - P_{I_j}[k+1] - P_{M_j}[k+1] \end{cases} \tag{3}$$

where $w_{i,j}$ are the edge weights, c_j is the 'click' probability of site j and $P_{I_i S_j}[k]$ is the joint probability that at time step k site i is infected, while site j is still susceptible. One may now think that using these equations it would be possible to solve the system numerically, but unfortunately the joint probabilities $P_{I_i S_j}$ are unknown, and there does not appear to exist an easy way to compute them exactly. One could simply resort to simulate the global Markov chain resulting from the model. However, simulations are very expensive in the case of a large system (millions of nodes), and it is hard to evaluate how many runs (and how long) are necessary to obtain a reliable prediction of the system behavior.

In the rest of this paper we will suggest bounds and approximations that can be used to quickly obtain gross-level predictions without resorting to simulations. To facilitate the reader we give some guidelines to follow the remainder of the paper. There are two main issues regarding the behavior of a malware infection that starts propagating in a network. The first one is the computation of the final average number of sites that will be infected. This is a problem in itself, which does not depend on the specific way in which we model the evolution of system. We will consider this problem separately in Section III. The second one is the derivation of the transient behavior of the system, that provides also the *temporal evolution* of

the average number of infected sites. In the long run this number tends to the limit resulting from the solution of the first problem, but the rate of convergence to the limit depends on how we formulate the dynamic interactions of the sites. Here is where our influence model comes into play. We will study the transient behavior in Section IV.

III. Percolation problem

In this section we explain why the problem of estimating the final number of infected sites reduces to what is known in physics as a *site percolation* problem, provided that the "click" probability is smaller than one. Then we will study the percolation problem on a simple case of small-world graph that illustrates the complexity of the analysis on an example which is particularly significant for e-mail virus propagation.

A. Reduction to a site percolation problem on a graph

The spreading of a new virus starts at time zero from a state in which a given set of sites are initially infected. Let I_0 be the initial number of infected sites. The propagation of the virus ends when there are no more contacts between an infected site and a susceptible one. Can we predict the final number of infected sites? If all of the sites are susceptible and their 'click' probabilities are equal to one, it is easy to understand that as time goes to infinity each site will receive a copy of the message containing the virus and will get infected. Now consider the general case in which a subset of the sites are initially immune, and those that are susceptible have a 'click' probability smaller than or equal to one. If the virus managed to reach every site on the graph, on average we would obtain a final number of infected sites $E[I_\infty] = I_0 + \sum_{i \in S} c_i$, where S is the set of initially susceptible sites. However, not all of the sites are necessarily reached by the virus. In fact, if the 'click' probability is sufficiently small, the virus reaches on average only a finite number of sites even in the case of an infinite number of susceptible sites. If we increase the value of the 'click' probability, assuming for simplicity that it is the same for each node, at a given point the system undergoes a phase transition that leads to the formation of a giant cluster of infected sites.

There exist a notion of "epidemic threshold" that is common to a wide variety of epidemic models regardless of the specific way in which the problem is formulated mathematically. The threshold usually refers to a single parameter of the model that describes the spreading capability of the infection. Below the threshold the expectation for the final number of infected sites is finite. Above the threshold, the final average number of infected sites goes to infinity (provided that there is an infinite number of susceptible sites). This fact is well known in the theory of random graphs [3]. The same phenomenon is known in physics as *site* or *bond percolation*, depending on whether the 'occupation probability' refers to the nodes or the edges of the graph. Our model of malware spreading maps precisely onto a site percolation problem, where the site occupation probability corresponds to the 'click' probability. Unfortunately, the exact solution of the site percolation problem is not feasible on a large graph arbitrarily connected, where it is necessary to resort to simulation. This is surely a major obstacle in studying analytically the problem of malware spreading. We could restrict the analysis to the most important case of a large infection outbreak, assuming that all of the susceptible sites will receive a copy of the virus. Under this assumption we should not care about the existence of a phase transition at all. However, if we want to study analytically infection processes that are below or close to the epidemic threshold, we have to solve the percolation problem.

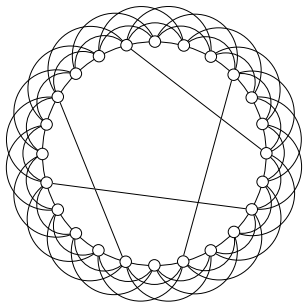

Fig. 2. Example of a small-world graph with $N = 24$, $k = 3$, $S = 4$

B. Site percolation on the small-world network model of Watts and Strogatz

As we already mentioned, the graph defined by email address books is expected to be a small-world graph. Recently, a simple model of small-world has been proposed by Watts and Strogatz [11] that has attracted the attention of many researcher, especially in the physics community, because it turns out to be amenable to treatment using a variety of techniques (see [10] for a survey of recent results).

The model consists of a regular lattice, in the simplest case a one-dimensional lattice with periodic boundary conditions, and a small number of 'shortcuts' bonds added between randomly chosen pairs of sites. More formally, we consider a graph of N sites arranged on a ring lattice, where each site is connected to its nearest neighbors up to some fixed range k, that we call lattice *connectivity*. Then, S additional links - the 'shortcuts' - are added between randomly selected pairs of sites (not already connected through the lattice). The *shortcut density* ϕ is defined as the ratio between the number of shortcuts and the number of links on the underlying lattice, so that $S = \phi k N$. An example with $N = 24$, $k = 3$ and $S = 4$ is shown on Figure 2. In the following we assume that the 'click' probability is the same for each node, equal to c, and that all of the nodes are initially susceptible.

Asymptotic results for the site percolation problem on this graph have been recently obtained by M. E. J. Newman using a generating function method [13], [14]. In [15] the same technique has been extended to the case of a two-dimentional lattice with shortcuts. These results are very useful, because they allow to determine exactly not only the epidemic threshold, but also the complete distribution of the sizes of infection outbreaks below the phase transition, as well as closed-form expressions for the mean and variance of the distribution. For example, the mean of the final number of infected sites in the

case of a one-dimensional lattice is given by (see [14])

$$E[I_\infty] = \frac{c\,(1+q)}{1-q-2k\phi c\,(1+q)} \tag{4}$$

where $q = 1-(1-c)^k$. Unfortunately, asymptotic results cannot be directly applied to our problem. Our goal is to study the temporal evolution of a malware infection that starts from a given set of initially infected nodes on a given (finite) topology. Asymptotic results provide only the final size of an infection originated from any initial node and averaged over every possible (infinite) realization of the small-world random graph model with parameters k, ϕ, c. Thus our problem is complementary to that considered by Newman.

We will now describe an algorithm that, given a realization of small-world graph over a one-dimensional lattice and the position of the initially infected node, addresses the problem of estimating the final probability (as time tends to infinity) that each node on the graph is reached by the virus. We call such a probability $P_R(i)$, where i is a progressive index for all of the nodes on the ring. We assign the index zero to the initially infected node. Note that $P_R(i)$ is just the probability of receiving a copy of the virus, not necessarily being infected by it. Using $P_R(i)$ the final average number of infected sites is given by

$$E[I_\infty] = 1 + \sum_{i=1}^{N-1} cP_R(i) \tag{5}$$

and can differ significantly from what is obtained using eq. (4), depending also on the position of the initially infected node.

A precise estimate of $P_R(i)$ can be obtained only in the case in which there are no shortcuts. The addition of shortcuts across the lattice leads to a problem belonging to complexity class NP, because the solution requires the consideration of all possible paths on the graph from the initially infected node to any other node, which increases exponentially with the number of shortcuts. We will first describe in Section III-B.1 the solution in the case of a pure lattice. Then in Section III-B.2 we will present a heuristic algorithm that obtains an upper bound, a lower bound, and a close approximation for the reaching probability in the general case.

1) Solution on one-dimensional lattice: The basic problem is to understand what occurs when a virus is injected into the lattice. Suppose for now that we have an infinite number of susceptible nodes arranged on a linear lattice with connectivity k. An example with $k = 3$ is shown on Figure 3.

An additional edge connects node 0 to another node (not represented), located outside the lattice, which we assume to be already infected. Node 0 will surely receive a copy of the virus, because it is directly connected to an already infected node. Hence, its reaching probability is one. If the virus infects node 0, which occurs with probability c, it starts an infection process within the lattice which reaches all of its neighbors up to a range k. Nodes at distance $k+1$ receive the virus if, besides node 0, at least one of the previous k nodes gets infected, leading to a reaching probability $b = cq$ (q has been already introduced in eq. [4]). As the distance from node 0 increases, it can be shown that the reaching probability

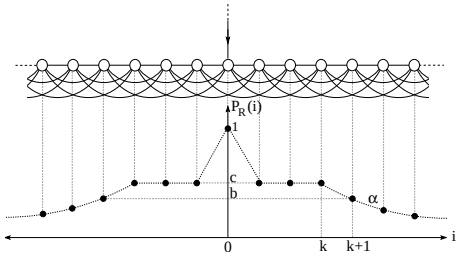

Fig. 3. Injection of virus into an infinite lattice with connectivity $k = 3$. The bottom part shows the reaching probability $P_R(i)$ as a function of the node index

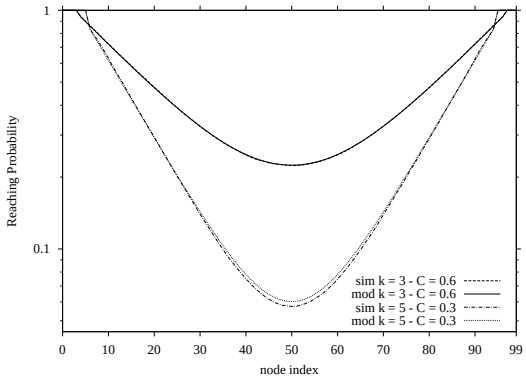

Fig. 4. Percolation on a ring lattice of $N = 100$ nodes with different choices of parameters k, c

decays geometrically on both sides of the infection origin. The parameter α of the geometric decay is given by (see [1])

$$\alpha = \frac{q - c\,(1-q)\,(k+q)}{q - kc\,(1-q)} \tag{6}$$

In the case of a finite number of nodes arranged on a ring lattice, each node can be reached by the virus from both sides of the infection origin. Letting P_R^1 and P_R^2 be the probabilities to be reached from one side or the other, the total reaching probability can be obtained combining these two probabilities in the following way:

$$P_R = 1 - (1 - P_R^1)\,(1 - P_R^2) \tag{7}$$

A comparison of results obtained from analysis and simulation in the case of $N = 100$, and the two combinations of parameters $\{k = 3\,,\, c = 0.6\}$ and $\{k = 5\,,\, c = 0.3\}$ is shown on Figure 4.

2) Bounds and approximations adding shortcuts: Now consider the case in which a given number of shortcuts are added to the ring lattice. The analysis is divided into two steps. In the first step we consider only the subset U of the nodes that includes the initially infected one and the vertices of the shortcuts. After having obtained the reaching probabilities of the nodes in U, the second step derives the reaching probabilities of all of the other nodes.

The first step works as follow: starting from the initially infected node we build a tree of the paths that can be followed by the virus during its propagation through the graph. Each path is an ordered list of edges whose vertices belong to U. Each edge e can be associated with a probability P_e that the virus traverses successfully that edge. While building the tree,

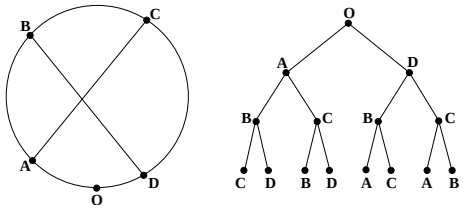

Fig. 5. Binary tree of the paths followed by the virus in the case of two shortcuts

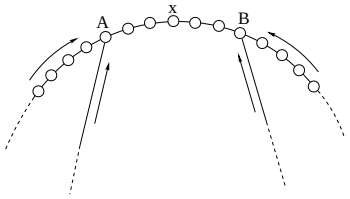

Fig. 6. Computation of the reaching probabilities of the nodes located between two vertices

we can compute for each path j the probability ${P_j}^i$ that the virus arrives at node i by multiplying the probabilities of the edges that have been traversed. The probability P_e depends on the type of edge. If it is a shortcut, the virus can use it to propagate across the network if both vertices get infected, which occurs with probability c^2. If the edge is a portion of the ring lattice, say from vertex A to vertex B, P_e depends not only on the distance d between A and B but also on way in which the infection arrives at A: if the virus arrives at A moving along the ring, we have $P_e = \alpha^d$. If the virus follows a shortcut to arrive at A, from what we said in Section III-B.1 we have $P_e = q\alpha^{(d-k-1)}$, if $d > k$, while $P_e = 1$ if $d \leq k$.

When a path encounters the vertex of a shortcut, it splits into two different paths, so that the resulting tree is binary. However a path cannot touch again an already visited vertex. An example for the case of two shortcuts in shown in Figure 5. The number of paths increases exponentially with the number of shortcuts, so that it is possible to consider all of them only with a limited number of shortcuts. Unfortunately, even if we are able to consider all of the paths, it is not possible to obtain the exact value of reaching probability of an arbitrary vertex.

Upper bound. We could combine the probabilities of all of the paths arriving at a vertex i as if they were independent:

$$P_R(i) = 1 - \prod_{j \in W_i} (1 - {P_j}^i) \tag{8}$$

where W_i is the set of distinct paths arriving at vertex i. This provides an upper bound to the reaching probability of node i, because two paths arriving at a vertex node may have in common some of the vertices that have already been visited, so that the probabilities ${P_j}^i$ are actually correlated.

Lower bound. It is possible to obtain a lower bound by applying the following method: we first consider the effect of the path arriving at i with the highest probability. Then we cancel this path from the tree, discarding all paths that share at least one edge with the removed path. We proceed considering the path with the highest probability among the surviving paths arriving at i, and so on, until there are no more paths from the origin of the infection to node i.

Improved bounds. It is possible to improve both the upper bound and the lower bound accounting for part of the correlations among the paths, in the following way. We compute the h most important edges in the tree followed by the paths arriving at vertex i [2]. Then we study separately all configurations of these h edges considered as a set of indipendent binary random variables (2^h combinations). Doing so we remove part of the correlations that were neglected before at the cost of a higher computational complexity for increasing values of h.

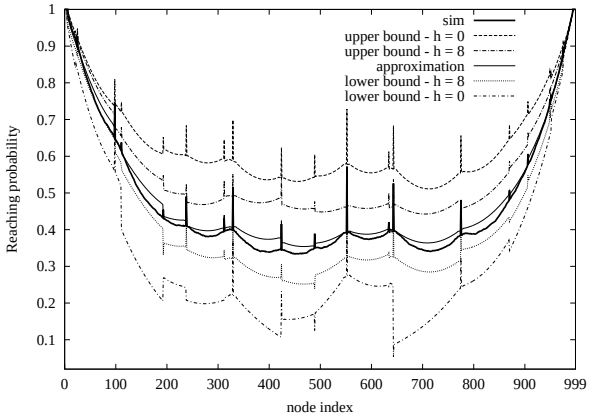

Fig. 7. Percolation problem adding 10 shortcuts to a lattice of 1000 nodes, $k = 5$, $C = 0.5$

Once we have obtained the reaching probability of the shortcut vertices, we can compute the reaching probabilities of the nodes located between two vertices. For each node X we consider all of the paths arriving at the nearest vertices A and B (see Figure 6) and combine their effect on X, which depends on the distance between A (or B) and X as well as on the type of path arriving at a vertex (either through a shortcut or along the ring). A comparison of results in the case of $N = 1000$, $k = 5$, $S = 10$, $C = 0.5$ is shown on Figure 7, which reports the reaching probability obtained averaging the results of 10000 simulations, the lower bound and the upper bound neglecting all path correlations ($h = 0$), and their improved versions using $h = 8$. These are indeed bounds because we were able to consider all of the paths in the tree. A good approximation is the mean between the lower and upper bounds obtained with $h = 8$, and it is reported on the plot. Note the peaks caused by virus injections into the lattice due to the shortcuts.

Unfortunately, if the number of shortcuts exceeds a few tens it is not possible to consider all of the paths in the tree, and different strategies could be adopted to face the computational

[2]One way to do this is to compute for each edge a sum of the probabilities ${P_j}^i$ of all of the paths arriving at i, and then sorting the edges on the basis of such sum

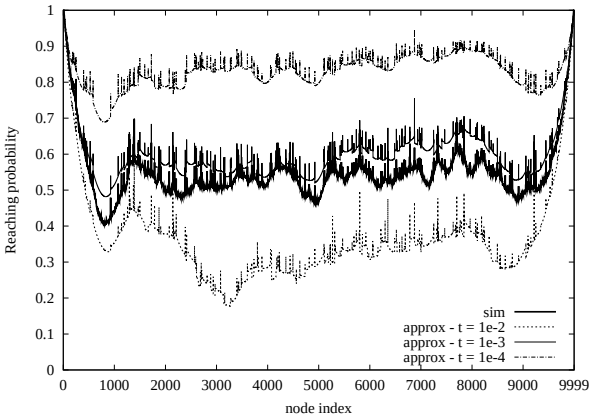

Fig. 8. Percolation problem adding 100 shortcuts to a lattice of 10000 nodes, $k = 10$, $C = 0.4$

complexity of the analysis [3]: one could make a "breadth first" search into the tree limiting the total number of paths to be computed, or a "depth first" exploration cutting the subtrees whose probability drops below a given threshold t. However, while it is still possible with this method to obtain lower bounds for the reaching probability (if we cut the tree we underestimate the spreading capability of the infection), it is not possible to derive a rigorous upper bound. After several experiments, we found that the best way to obtain a quick, rough approximation of the reaching probability (but not a bound) is to use the algorithm to compute the upper bound making a depth first analysis up to a suitable threshold t. An example of results in the case of $N = 10000$, $k = 10$, $S = 100$, $C = 0.4$ is shown on Figure 8, that reports the average of 1000 simulation experiments (that took about six hours on our machine), together with analytical results obtained using three different cutting thresholds (the entire analysis took less than one minute). The approximation obtained using $t = 10^{-3}$ is quite accurate, however it is unclear whether it is possible to choose a priori a suitable value of the threshold, based on the system parameters, in order to obtain a reliable estimate.

Finally, it is interesting to observe in Figure 8 that, away from the origin of the infection, the reaching probability is almost the same for a large fraction of the nodes (about 0.5). This behavior suggests that the critical phase of the spreading of a virus is the very beginning of the infection: if the virus manages to conquer a few strategic points around the origin (the vertices of the nearest shortcuts) then it is likely to reach all of the other nodes on the graph.

IV. Transient analysis

Our primary concern in modeling malware spreading dynamics is to understand the temporal evolution of a new infection as it starts propagating in a network. So far in the literature this kind of analysis has been carried out only neglecting the impact of the underlying topology, or resorting to simulation. A simplification that has been adopted is to assume that each node is equally likely to be infected by any other node on the graph (this is usually called "homogeneous" assumption), but this is clearly a rough approach, perhaps acceptable only for certain kinds of malware, such as worms that propagates performing a random scanning of the IP address space, like Code Red.

Although the exact analysis in the case of an arbitrary topology appears to be unfeasible, using our stochastic model based on Interactive Markov Chains it is possible to obtain at least some useful bounds, as well as satisfactory approximations, as we will show in this Section.

To avoid the additional complexity introduced by the percolation phenomenon, we will first consider in Section IV-A the case in which the 'click' probability is equal to one. This implies that, after the transient phase corresponding to the spreading of the virus, the system settles down to a final configuration in which all of the nodes are infected. The main goal of the analysis, in this case, is to determine how long does it take to the virus to infect all of the nodes starting from an arbitrary point. We will observe that the topology of the graph plays a crucial role that can be predicted analytically, getting interesting insights into the behavior of malware on a network. The transient analysis will be extended to the case of 'click' probability smaller than one in Section IV-B.

A. The case of 'click' probability equal to one

We already described in Section II-B the recursive equations (3) that allow one to solve numerically the state evolution of the system. The major problem is that we do not know how to compute the joint probabilities $P_{I_iS_j}[k]$ of pairs of neighboring nodes. However, we can establish simple lower and upper bounds for such joint probabilities, for arbitrary values of click probabilities, as follows. We introduce $P_{R_i}[k] = P_{I_i}[k] + P_{M_i}[k]$, the probability that site i has been already reached by the virus at time k, being infected or immune respectively with probability c_i and $1 - c_i$.

Lower bound. We rewrite $P_{I_iS_j}[k]$ as

$$P_{I_iS_j}[k] = P_{I_i}[k] - P_{I_iR_j}[k]$$

Exploiting the property of joint probabilities

$$P_{I_iR_j}[k] \leq \min\left(P_{I_i}[k], P_{R_j}[k]\right)$$

we obtain

$$P_{I_iS_j}[k] \geq P_{I_i}[k] - \min\left(P_{I_i}[k], P_{R_j}[k]\right) \qquad (9)$$

According to equations (3), this leads to a lower bound for the infection probability of node i at step $k+1$, which is an increasing function of each pair of joint probabilities $P_{I_iS_j}[k]$. Using this bound on any node at any time, we obtain a lower bound for the entire infection process on the graph.

Upper bound. An upper bound can be obtained from the fact that $P_{I_i}[k]$ and $P_{S_i}[k]$ are negatively correlated. To prove that, we consider the binary random variables $R_i[k]$, equal to 1 if node i at time k has already been reached by the virus; $R_i[k]$ is instead equal to 0 if node i has not yet been reached by the virus.

[3] it becomes quite expensive also to obtain accurate results by simulation, because each simulation run represents only one of a wide variety of realizations of the same infection process on the graph

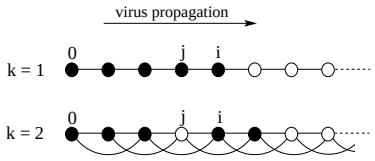

Fig. 9. Examples of possible states of the system in the case of a lattice with $k = 1$ (upper part) and $k = 2$ (lower part)

We can easily recognize that $R_i[k]$ and $R_j[k]$ (where nodes i and j are neighbors) are *associated random variables* in the sense of [2]. From the theory of [2] we have that

$$P_{R_i R_j}[k] \geq P_{R_i}[k] \cdot P_{R_j}[k]$$

Since $P_{I_i}[k] = c_i P_{R_i}[k]$ and $P_{S_j}[k] = 1 - P_{R_j}[k]$, it follows that $P_{I_i}[k]$ and $P_{S_i}[k]$ are negatively correlated, that is

$$P_{I_i S_j}[k] \leq P_{I_i}[k] \cdot P_{S_j}[k] \tag{10}$$

which provides the desired upper bound for joint probabilities $P_{I_i S_j}[k]$.

In words, if we assume that the status probabilities $P_{I_i}[k]$ and $P_{S_j}[k]$ at any given time are independent for each pair of neighboring nodes, we overestimate the spreading rate of the infection. These probabilities are instead negatively correlated, and this correlation has an important impact on the state evolution described by equations (3).

The amount of correlation that arises in the statuses of neighboring nodes strongly depends on the underlying structure of the graph. The impact of topology can be easily shown in the simple case of the infinite one-dimensional lattice already considered in Section III-B.1. Figure 9 shows examples of possible states of the lattice at a certain time instant, for two different values of connectivity k. Black circles represents infected nodes, while white circles represents susceptible nodes not already reached by the virus. The initial infected node is at index 0, and the lattice extends indefinitely to the right for increasing values of the index. As time goes on, the virus propagates to the right infecting all of the nodes. If we consider the pair of neighboring nodes i and j, we can easily prove that the lower bound (9) is exact in the case of $k = 1$, that is to say $P_{R_i R_j}[k] = \min(P_{R_i}[k], P_{R_j}[k])$. This comes from the fact that the farthest node, i , can be infected only if the nearest node, j , has already been infected, so that the conditional probability $P_{R_j|R_i}[k]$ is equal to one. We observe that this holds if and only if there is a unique path from the origin of the infection to any other node. Therefore, the infection process can be solved exactly on all graphs that exhibit a tree structure. However, as soon as we increase the connectivity of the graph, the above equality does not hold anymore. The lower part of Figure 9 shows an example with $k = 2$ in which we have a 'hole' at position j, so that $P_{R_j|R_i}[k] < 1$. Correlations become weaker as we increase the connectivity of the lattice.

A comparison of results obtained from simulation and analysis (upper bound and lower bound) is shown on Figure 10 for three different values of connectivity k. The graph

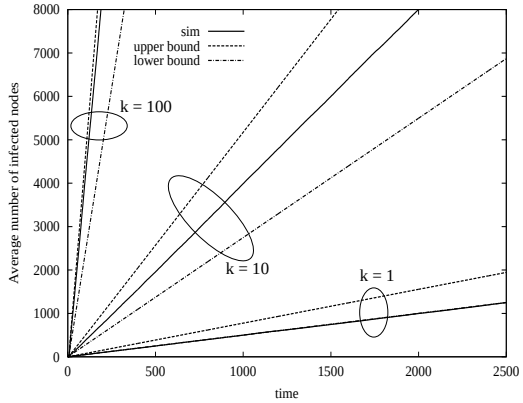

Fig. 10. Comparison of results obtained by simulation and analysis for $k = 1$, $k = 10$ and $k = 100$ on the one-dimensional lattice

reports the evolution of the average number of infected sites in the infinite lattice, as a function of time. Note that for $k = 1$ the lower bound provides the exact result, while for $k = 100$ the simulation curve is much closer to the upper bound.

Approximation. Besides the bounds, that hold on any type of graph, we found that it is possible to compute an accurate approximation of the state evolution on the one-dimensional lattice by means of a simple mixture of the upper bound **UB** and lower bound **LB** defined by equations (10) and (9), respectively.

$$P_{I_i S_j} = (M)\ \mathbf{UB} + (1 - M)\ \mathbf{LB} \tag{11}$$

where M is a suitable mixing coefficient, that depends not only on k, but also on the probability s that a node gets influenced by itself. In fact the larger the weight associated with the self-loop of a site, the more independent is the status of that site from the statuses of its neighbors. On a one-dimensional lattice of 10000 nodes, we explored by simulation a wide range of values of k and four values of s (0 , $1/3$, $2/3$, 0.9) and we derived the mixing coefficient M that yields the most accurate approximation for the evolution of the number of infected nodes. We obtained the points shown on Figure 11, and we found that they are well fit by the following empirical function:

$$M(k, s) = \frac{1}{1 + \frac{a_1 + a_2 s}{\ln[(a_2 s + a_3)k + a_5]}} \tag{12}$$

where $a_1 \cdots a_5$ are parameters computed by a fitting procedure based on a non-linear least-squares algorithm. From our results it seems that the independence assumption ($M = 1$) holds only in the limit as $k \to \infty$, and M increases roughly linearly only with the logarithm of k. The dependence on s is weaker, but still significant. On a regular lattice, M is the same for all pairs of neighboring nodes. Our proposed solution to deal with a general topology is to use the same formula (12) derived on the one-dimensional lattice, but substituting the local connectivity and the self-influence of the node that gets influenced. That means that we compute a coefficient $M(k_j, s_j)$ for each node on the graph. In synthesis, in order to approximate the joint probabilities $P_{I_i S_j}$ necessary to compute the status evolution of node j (equations [3]), we use the

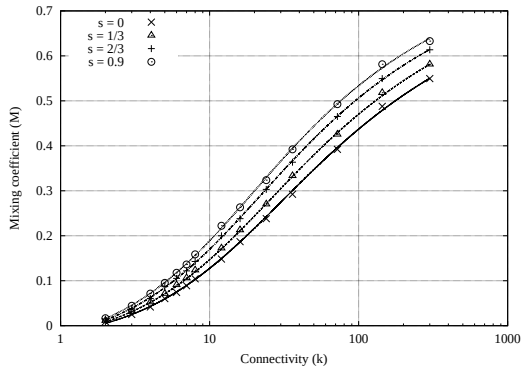

Fig. 11. Mixing coefficients obtained from simulation as a function of k (horizontal axis) and s (parameter), and empirical curves used to fit the simulation points

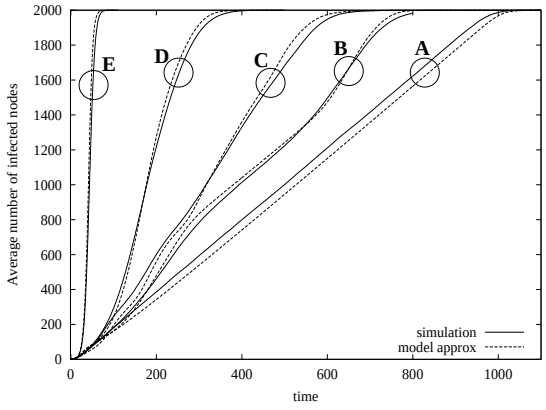

Fig. 12. Comparison of results obtained by simulation and approximate analysis on a network of 2000 nodes

mixture of bounds (11) where M is obtained by formula (12) plugging in the local connectivity k_j and the self-influence s_j of node j. To be consistent with the meaning of connectivity on the regular lattice, we define $k_j \equiv n_j/2$, where n_j is the number of neighbors of node j. We found that this approximation produces satisfactory results also on irregular topologies. Figure 12 plots the evolution of the average number of infected nodes on a ring lattice of 2000 nodes in five different cases that, besides validating our approximation, provide also interesting insights into the dynamics of virus propagation:

- **A** - regular lattice, $k = 10$, $s = 0.8$, homogeneous weights $w = 0.01$
- **B** - same lattice as A, adding two shortcuts between nodes $50 - 1600$ and $500 - 900$
- **C** - lattice with variable k_j taken from a geometric distribution with a mean of 10, truncated at 50, homogeneous weights $w = 0.01$
- **D** - same lattice as A, adding 20 shortcuts
- **E** - fully connected graph, $s = 0.8$, homogeneous weights

Comparing **A** with **B**, we observe the impact of the two shortcuts, and the accuracy of the model in predicting the

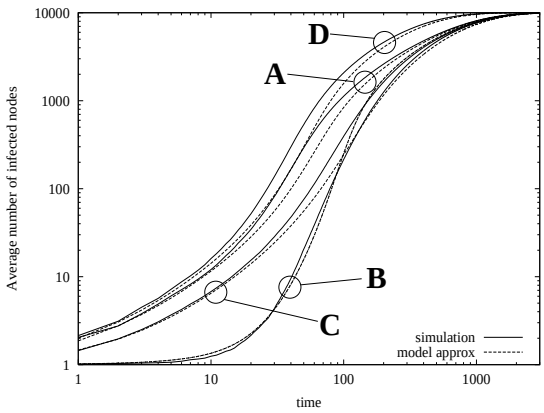

Fig. 13. Comparison of results obtained by simulation and approximate analysis on power law graphs generated with the GLP algorithm

points in which the spreading rate changes [4]. **C** shows that the model is accurate also in the case of an irregular lattice. Moreover, comparing **C** with **A**, we draw the interesting conclusion (confirmed by other experiments not shown here) that an increase in the variance of the local connectivity (while maintaining the same average) significantly raises the spreading rate of the infection. **D** shows again the dramatic effect of the shortcuts, since the addition of just 40 edges (20 bidirectional shortcuts) on a graph with 40000 edges (2000 nodes, each with 20 incoming edges) alters the shape of the curve making it much more similar to an exponential growth than the linear growth exhibited by **A**. Finally, **E** shows how faster is the propagation of the virus on a fully connected graph with respect to the other cases.

The approximate analysis proves to be accurate also on topologies much different from the small-world model of Watts and Strogatz. We used the BRITE topology generator[5] to build power-law graphs of 10000 nodes employing the GLP algorithm described in [16]. A power-law graph is defined as a graph in which the ccdf of the node degree d satisfies $F(d) \propto d^{\alpha}$, $\alpha < 0$. We adopted the GLP algorithm, that was designed to match the power low exponent and the clustering behavior of the AS-level Internet topology, just in order to generate topologies very different from a ring lattice. The graphs were generated using a constant value 0.45 for the parameter p of GLP algoritm (see [16]), while trying different values of m, which is the initial connectivity of the nodes as they are added to the graph during the generation process. Figure 13 plots on a log-log scale the evolution of the average number of infected nodes in four different cases, comparing results obtained from both simulation and approximate analysis:

- **A** - $\alpha = -1.14$, $m = 1$, infection origin on the node with the highest degree ($d = 510$)
- **B** - same graph as **A**, infection origin on a node with degree $d = 10$

[4]It would be possible to explain the shape of the curve by means of a few observations based on the position of the shortcuts and the linear propagation of the virus on the ring lattice

[5]BRITE is available at `http://cs-pub.bu.edu/brite/`

- **C** - $\alpha = -1.6$, $m = 1$, infection origin on the node with the highest degree ($d = 354$)
- **D** - $\alpha = -1.14$, $m = 2$, infection origin on the node with the highest degree ($d = 586$)

To obtain a meaningful comparison, all of the weights associated with the edges are identical across all four cases, equal to 0.017. The distance between **A** with **B**, that were obtained exactly on the same graph, shows that the position of the infection origin indeed plays a significant role. When the virus originates from a low degree node, we observe a delay in the start-up of the infection that is due to the fact that the virus needs some time before reaching the core of the network. **C** was obtained on a different graph with the same number of edges but with a node degree distribution less heavy-tailed than that relative to **A**. The infection was again started on the node with the highest degree. We observe again the effect already shown on Figure 12, that is to say the higher the variance of the node degree, the faster the spreading of the virus. Finally, **D** refers to a graph with the same power law exponent of **A**, but with double the number of edges. Increasing the connectivity of the graph always accelerates the spreading of the virus. In fact it is possible to show easily (see [1]) that the addition of any edge to a given graph (without changing the probabilities associated with the other edges) makes the infection spread faster. The same property does not hold if, adding the edge, we repartition the weights associated with the pre-existing edges, as can be shown with a simple counter-example.

The results from the approximate analysis follows closely the curve derived from simulations in all four cases. We have not shown the bounds (9) and (10). It is worth noting that on this kind of random graphs the upper bound is much closer to simulation than the lower bound, so that we conclude that correlations between adjacent nodes are rather weak.

B. The case of 'click' probability smaller than one

In this Section we extend the transient analysis to the case of 'click' probability smaller than one. This requires us to first solve the percolation problem described in Section III, in case we want to analyze a system below the epidemic threshold. Otherwise, we can assume that as time tends to infinity all of the nodes will be reached by the virus. We assume for now that we have an estimate of the reaching probability of the nodes, $P_R(i)$, for every node on the graph. Our proposed solution to derive the state evolution of the system is quite simple, although not rigorously correct. We simply use equations (3) letting $S_i[0] = P_R(i)$ and $M_i[0] = 1 - P_R(i)$. In words, a node with a reaching probability $P_R(i)$ is considered already immune at time 0 with probability equal to the probability that it is not reached by the virus, which is the complement of $P_R(i)$. This way, only the reaching probability, assigned to the probability to be initially susceptible, is going to be partitioned into the final probability to be infected - equal to $c_i P_R(i)$ - and an additional probability to become immune because the user does not click on the attachment containing the virus - equal to $(1 - c_i)P_R(i)$. Using this approach the average final number of infected nodes is correct. However, we overestimate the spreading rate of the infection, as will be seen on an experiment performed on the small-world graph. We consider 1000 nodes on a ring lattice of connectivity $k = 6$, and we add 4 shortcuts (precisely those connecting the pairs of nodes 198-760, 525-94, 276-211, 542-997). We compute an upper bound of $P_R(i)$ using the algorithm described in Section III-B.2 with $h = 12$, that is quite close to the exact reaching probability derived from simulation, and we perform the transient analysis as described in Section IV-A with the position $M_i[0] = 1 - P_R(i)$. Varying the 'click' probability from 1 to 0.3, we obtained the results shown on Figure 14. The approximate analysis, which is very accurate in the case of $c = 1$, tends to overestimate the spreading rate of the virus especially near the percolation transition, which occurs for a 'click' probability between 0.7 and 0.6. This error can be interpreted as follows: letting $S_i[0] = P_R(i)$, we make all of the 'reachability' of a node already available at time 0, while this is not correct, because $P_R(i)$ is the result of the superposition of infection processes that follows different paths over the graph, arriving at node i at different time instants. Actually, $P_R(i)$ should be an increasing function of time. Our simple solution thus overestimates the average number of nodes that can be infected at a given time. Note that a rigorous upper bound is obtained only by combining the initial condition $S_i[0] = P_R(i)$, where $P_R(i)$ is itself an upper bound of the final reaching probability, with the upper bound for joint probabilities (10). On the other end, a lower bound seems to be more complicated to be obtained.

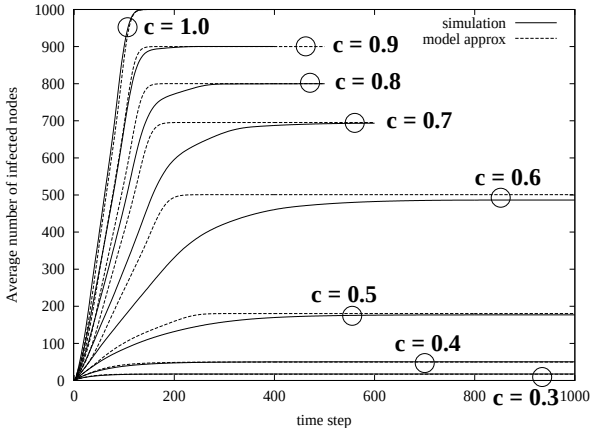

Fig. 14. Transient analysis on a small-world graph of 1000 nodes for different values of 'click' probability

We considered also the case of a 'click' probability smaller than one on more general topologies than the small-world graph. On the same power-law graph built in Section IV-A (more precisely the one used to derive the case **A** of figure 13) we started an infection process on the node with the highest degree, obtaining the curves shown on Figure 15 for different values of c (constant on the graph). The plot compares simulation results with two types of analysis: 'model approx' solves equations (3) from the initial condition in which we let $S_i[0] = 1$. We observe that, according to this model, all of the nodes on the graph tend to be reached by the virus, since the final number of infected nodes approaches cN. The model 'approx + bound perc', instead, accounts in a simple way for the percolation phenomenon that arises on the graph.

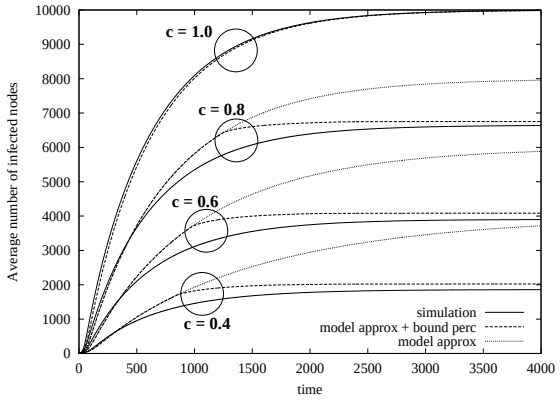

Fig. 15. Comparison of results obtained by simulation and approximate analysis on a GLP random graph, for different values of 'click' probability

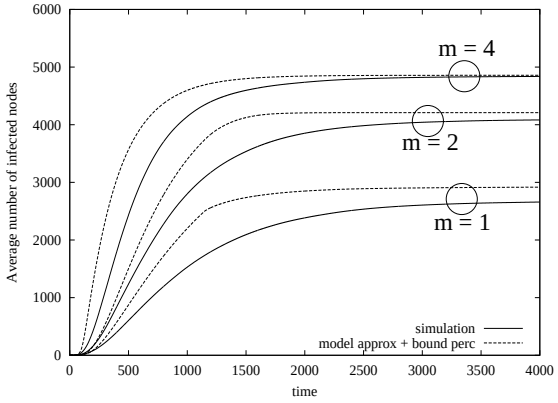

Fig. 16. Comparison of results obtained by simulation and approximate analysis on GLP graphs generated with different m, and constant $c = 0.5$

As already mentioned before, the solution of the percolation problem is tough on an arbitrary topology, but a very simple upper bound of the reaching probability of any node i can be obtained assuming that the node is *not* reached by the virus only if all of its neighbors do not 'click':

$$P_R(i) = 1 - \prod_{j \in n_i} (1 - c_j) \tag{13}$$

where n_i is the number of neighbors of node i [6]. Figure 15 shows that this bound is quite close to the actual result obtained by simulation. This is also due to the fact that a large number of nodes are attached to the network only by a single edge, and for these nodes the bound (13) is exact, because they cannot be infected if the node to which they are connected does not get infected as well.

Being quite unusual that an email address has a single contact with another email address, we changed the parameter m of GLP algorithm that sets the initial degree of nodes that are added to the graph. Moreover, we considered the case in which the infection starts on a node with degree 10, and we used a constant 'click probability' equal to 0.5. Results are shown on Figure 16 for different values of m. We observe that the upper bound (13) is still accurate if we increase the node connectivity. Moreover, the final number of infected nodes approaches the upper limit cN already using $m = 4$. This means that in this case the number of different paths connecting the initial infected node to any other node is so huge that in the long run all of the nodes are reached by the virus. The model, however, overestimates the spreading rate of the virus, mainly because it assumes that $S_i[0] = P_R(i)$, causing the deviation already pointed out earlier in this Section.

V. Open issues and further work

In this Section we briefly report on the main modeling issues left open in our work, suggesting directions for further research. As far as the percolation problem is concerned, one could refine the solution on the small-world network model of Watts and Strogatz, perhaps investigating the case of irregular connectivity or non-homogeneous 'click' probability, or even try to extend the approach to the case of a two-dimensional lattice with shortcuts, but it is unclear if this simplified model indeed applies to the graph defined by email contacts, whose properties are still unknown. As far as the transient analysis is concerned, it would be desirable to better understand the nature of correlations between adjacent nodes, justifying analytically the shape of the curve relative to the mixing coefficient introduced in Section IV-A, which has been derived only empirically. Finally, preliminary results not reported here (see [1]) show that many effects due to the network structure can be explained looking at the eigenvalues of the influence matrix H introduced in Section II-A.

[6] This formula does not apply to the nodes that are directly connected to the initial infected node, because they are surely reached by the virus

VI. Conclusion

In this paper we presented an analytical framework, based on *Interactive Markov Chains*, that can be used to study the dynamics of malware propagation on a network. The exact solution of a stochastic model intended to capture the probabilistic nature of malware propagation on an arbitrary topology appears to be a major challenge, because of the high computational complexity necessary to analyze very large systems. However, one can resort to simple bounds and approximations in order to obtain a gross-level prediction of the system behavior that can help to understand important characteristics of malware propagation. Although we have focused on the modeling aspects of the problem, we believe our methodology can be usefully applied to evaluate different countermeasures against future malware activity, as well as fundamental issues on network vulnerability assessment. Moreover, the flexibility of the approach based on IMCs allows to apply our work beyond the problem of malware spreading, addressing a wide variety of dynamic interactions on networks. Our modeling effort is to be considered a first step in a rather novel research area that we expect to gain more and more relevance in the next future.

References

[1] M. Garetto, "Modeling Malware Spreading Dynamics," extended version, `http://www1.tlc.polito.it/~garetto/pub/virusreport.ps.gz`

[2] R. E. Barlow, F. Proschan, "Statistical Theory of Reliability and Life Testing," Holt, Rinehart and Winston, Inc., New York, 1975

[3] P. Erdös and A. Rényi "On the Evolution of Random Graphs," *Publ. Math. Inst. Hungar. Acad. Sci,* **5**, 17-61, 1960

[4] J. O. Kephart ans S. R. White, "Directed -graph Epidemiological Models of Computer Viruses", *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy,* pp. 343-359,1991.

[5] H. Andersson, T. Britton, "Stochastic Epidemic Models and Their Statistical Analysis", Lecture Notes in Statistics, Springer-Verlag, 151, (2000)

[6] C. Wang, J. C. Knight, M. C. Elder,"On Computer Viral Infection and the effect of Immunization," *in Proc. 16th ACSAC,* 11–15 December, New Orleans, Louisiana, 2000

[7] D. L. Pepyne, C. G. Panayiotou, C. G. Cassandras and Y. C. Ho, "Vulnerability Assessment and Allocation of Protection Resources in Power Systems," *in Proc. of American Control Conference,* pp. 4705–4710, Arlington, VA, June 2001

[8] D. L. Pepyne, W. B. Gong and Y. C. Ho, "Modeling and Simulation for Network Vulnerability Assessment," *40th U.S. Army Operation Research Symposium (AORS XL),* Fort Lee, VA, October 2001

[9] C. Asavathiratham, "Influence Model: A tractable Representation of Networked Markov Chains," `http://tanzeem.www.media.mit.edu/people/tanzeem/cohn/chalee-thesis.pdf`

[10] M. E. J. Newman "Models of the small world", *J. Stat. Phys*, 101, 819-841 (2000).

[11] D. J. Watts, S. H. Strogatz, "Collective dynamics of 'small-world' network," *Nature* 393, 440-442 (1998)

[12] K. Houle, G. Weaver,"Trends in Denial of Service Attack Technology", `http://www.cert.org/`, October 2001

[13] C. Moore and M. E. J. Newman "Epidemics and percolation in small-world networks," *Phys. Rev. E* **61**, (2000)

[14] C. Moore and M. E. J. Newman, "Exact solution of site and bond percolation on small-world networks,", *Phys. Rev. E* **62**, (2000)

[15] M. E. J. Newman, I. Jensen, and R. M. Ziff, "Percolation and epidemics in a two-dimensional small world," *Phys. Rev. E* **65**, (2002)

[16] T. Bu and D. Towsley,"On Distinguishing between Internet Power Law Topology Generators," *in Proc. Infocom 2002,* June 23–27, New York.