Modelling and Simulating the Propagation of Computer Worms

Xiang Fan

Doctor of Philosophy

CQUniversity Australia

School of Engineering and Technology

July 2013

CERTIFICATE OF AUTHORSHIP AND ORIGINALITY OF THESIS (DECLARATION)

The work contained in this thesis has not been previously submitted either in whole or in part for a degree at CQUniversity or any other tertiary institution. To the best of my knowledge and belief, the material presented in this thesis is original except where due reference is made in text.

Signed:

Date: 30 July 2013

COPYRIGHT STATEMENT

This thesis may be freely copied and distributed for private use and study, however, no part of this thesis or the information contained therein may be included in or referred to in publication without prior written permission of the author and/or any reference fully acknowledged.

Signed:

Date: 30 July 2013

Abstract

Active worms propagate across networks by employing various target discovery techniques. It is anticipated that a future active worm would employ multiple target discovery techniques simultaneously to greatly accelerate its propagation. Strategies that future active worms might employ to shorten the slow start phase in their propagation are studied. Their respective cost-effectiveness is assessed.

This thesis also presents a study on modelling and simulating the propagation of Peer-to-Peer (P2P) worms. Motivated by the aspiration to invent an easy-to-employ instrument for research on the propagation of P2P worms, I model the propagation processes of P2P worms by difference equations of logic matrix, which are essentially discrete-time deterministic propagation models of P2P worms. To the best of my knowledge, I am the first using logic matrix in network security research. The instrument's ease of employment, which is demonstrated by its applications in our simulation experiments, makes it an attractive tool to conduct research on the propagation of P2P worms.

The major contributions in this thesis are firstly, the combination of target discovery techniques that can best accelerate propagation of active worms was suggested; secondly, strategies to shorten an active worm's slow start phase in its propagation were assessed based on a cost and benefit analysis; thirdly, I proposed a novel logic matrix approach to modelling the propagation of P2P worms; and fourthly, I found the impacts of the two different topologies on a P2P worm's attack performance, and compared the effects of two different quarantine tactics.

Table of Contents

List of Tables

List of Figures

\

Acknowledgments

First and foremost, I am greatly thankful to my wife, Ko Hyeon Young, for her patient care and love.

I would like to express my sincere thanks to my principal supervisor Professor William Guo. Without the invaluable advice, encouragement and support from him, this thesis would not have become an existence. Much of what lies in this thesis can be credited to his patient advice and persistent encouragement.

My thanks are also given to my associate supervisor Professor Mark Looi.

Finally, I thank my previous supervisors, Professor Yang Xiang and Dr Shawkat Ali, for their supervision in the early stage of my PhD study.

List of Publications

X. Fan, W. W. Guo, and M. Looi, "Modeling and simulating the propagation of structured peer-to-peer worms," *Journal of Networks*. **(Accepted) (ERA Tier A Journal)**

X. Fan, W. W. Guo, and M. Looi, "Modeling and simulating the propagation of unstructured peer-to-peer worms," in *2011 Seventh International Conference on Computational Intelligence and Security*, Sanya, China: Los Alamitos, California.: IEEE Computer Society, 2011, pp. 573-577.

X. Fan and Y. Xiang, "Modeling the propagation of peer-to-peer worms," *Future generation computer systems*, vol. 26, (2010), pp. 1433-1443, 2010. **(ERA Tier A Journal)**

Y. Xiang, **X. Fan**, and W. T. Zhu, "Propagation of active worms: a survey," *International journal of computer systems science & engineering*, vol. 24, pp. 157-172, 2009. **(ERA Tier C Journal)**

X. Fan and Y. Xiang, "Accelerating the propagation of active worms by employing multiple target discovery techniques," in *Lecture notes in computer science*. vol. 5245/2008 J. C. e. al, Ed. Berlin: Springer Verlag, 2008, pp. 150 -161.

X. Fan and Y. Xiang, "Shortening the slow start phase in the propagation of active worms," in *CSA 2008: Proceedings of International Symposium on Computer Science and Its Applications*, Hobart, Australia: Los Alamitos, California.: IEEE Computer Society, 2008, pp. 90-95.

Additional Publications by the Candidate

X. Fan and Y. Xiang, "Defending against the propagation of active worms," in *Proceedings of the 5th International Conference on Embedded and Ubiquitous Computing,* Shanghai, China: Los Alamitos, California.: IEEE Computer Society, 2008, pp. 350-355.

X. Fan and Y. Xiang, "Defending against the propagation of active worms," *Journal of supercomputing,* vol. 51, pp. 167-200, 2010. **(ERA Tier B Journal)**

X. Fan and Y. Xiang, "Modeling the propagation process of topology-aware worms," in *Network and parallel computing: 2009 Sixth IFIP International Conference on Network and Parallel Computing (NPC 2009),* Gold Coast, Australia: Los Alamitos, California.: IEEE Computer Society, 2009, pp. 182-189.

X. Fan and Y. Xiang, "Propagation modeling of peer-to-peer worms," in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA),* Perth, Australia: Los Alamitos, California: IEEE Computer Society, 2010, pp. 1128-1135.

X. Fan and Y. Xiang, "Modeling the propagation of peer-to-peer worms under quarantine," in *2010 IEEE/IFIP Network Operations and Management Symposium 2010,* Osaka, Japan: Piscataway, NJ: IEEE, 2010, pp. 942-945.

Chapter 1 Introduction

1.1 Background

A computer worm is ‘a program that self-propagates across a network exploiting security or policy flaws in widely-used services’ [1]. In order to spread, computer worms need to discover hosts with a certain particular vulnerability by employing some target discovery techniques.

Computer worms employing topological scanning as their target discovery technique are called topology-aware worms. Typical examples of topology-aware worms are worms attacking a flaw in a Peer-to-Peer (P2P) application and propagating across the P2P network by getting lists of peers from their victims and directing their subsequent attacks to those peers. This sort of topology-aware worms are called P2P worms.

In recent years, computer worms have been one of the most serious threats to current Internet infrastructure due to their rapid propagation, causing huge amount of financial loss and social disruption. In order to find an effective defense mechanism to curb the rapid propagation of computer worms, we must study their propagation mechanisms thoroughly because I believe only by fully understanding the attack mechanisms can we perform effective and comprehensive defence [2, 3].

In this thesis, propagation mechanisms of computer worms are studied systematically. Due to the crucial role that target discovery techniques play in determining the propagation characteristics of computer worms, a thorough study of the various target discovery techniques is imperative if we want to have a deep insight into how to find an effective solution to the rapid propagation of computer worms across the Internet.

1.2 Aim of the Research and Justification

The aim of this research is to establish mathematical models of computer worms, especially P2P worms.

All of the existing models are not applicable to computer worms employing topological scanning. The proposed models are suitable for modeling P2P worms because these models take into account topology of a P2P network.

1.3 Methodology

There are several different ways to study the characteristics of a piece of self-propagating code. The most powerful approach is probably the creation of realistic mathematical models that allow behavior prediction in a closed form. The problem with this approach is that such models are not generally available and are usually hard or even impossible to create. In a sense simulation is a mathematical model in which some of the functions used rely heavily on iteration. In order to reduce computational complexity, abstraction and approximation of the inner mechanisms of the object studied are often used. This allows computation of functions that are not well understood in a mathematical sense. Here, the analytical approach of mathematical modeling is replaced with an experimental approach, in which scenarios are simulated and then analysed. Simulation experiments are often very effective tools to understand complex processes.

For the proposed research, I will first establish mathematical models of computer worms. Then, I will conduct simulation experiments based on the mathematical models established. Finally, results from the simulation experiments will be analysed to draw conclusions.

1.4 Scope of the Research

Chapters 3 and 4 are dedicated to investigation of non-P2P worm propagation. In chapters 5 and 6, logic matrix representation is used for investigation of P2P worm propagation. Since these 2 kinds of computer worms are fundamentally different in terms of their respective propagation mechanism, which has been covered in the thesis, I treat them separately. Therefore, how to employ the combination of target discovery techniques to accelerate P2P worm propagation using the logic matrix representation is outside the scope of this research.

1.5 Major Contributions of the Thesis

The major contributions of this thesis are as follows.

- It was found that uniform scanning is an indispensable elementary target discovery technique of active worms. This point is of extreme importance when multiple target discovery techniques are to be employed, which means uniform scanning must be included as one of those target discovery techniques to be employed.
- I found the combination of target discovery techniques that can best accelerate the propagation of active worms.

- I proposed a discrete-time deterministic CFAP model of active worms.
- Ii was derived from mathematical analysis that in order to accelerate an active worm's propagation, I must try to let the active worm infect the first susceptible hosts and enter its fast spread phase as soon as possible. This point gives guidance to how to best accelerate an active worm's propagation.
- I proposed several strategies to shorten an active worm's slow start phase in its propagation, and found the cost-effective hit-list size and average size of internally generated target lists based on our cost and benefit analysis.
- I proposed a novel logic matrix approach to modelling the propagation of P2P worms by modelling the propagation processes of P2P worms by difference equations of logic matrix.
- I found the impacts of the two different topologies, namely structured and unstructured P2P networks, on a P2P worm's attack performance; and compared the effects of two different quarantine tactics, namely random quarantine and priority quarantine.

1.6 Structure of the Thesis

This thesis is organized as follows.

- Chapter 2 surveys related work, which sets the stage for later chapters.
- Chapter 3 examines the impact of employing multiple target discovery techniques simultaneously on the propagation characteristics of active worms.

- Chapter 4 investigates the impact of shortening the slow start phase in the propagation of active worms on their propagation characteristics.
- Chapter 5 derives from first principle an innovative logic matrix formulation of the propagation process of P2P worms under three different conditions [4, 5].
- Chapter 6 applies the logic matrix approach in simulation experiments [6-8].
- Finally, Chapter 7 concludes this thesis.

Chapter 2 Related Work

Computer worms can be classified according to the techniques by which they discover new targets to infect: scanning, pre-generated target list, internally generated target lists, or passive monitoring [1]. Active worms choose only the first three target discovery techniques.

2.1 Target Discovery Techniques of Active Worms

As mentioned in the previous section, active worms seek out victim hosts by employing such target discovery techniques as scanning, pre-generated target list, or internally generated target lists. In this section, I discuss these target discovery techniques and different types of them, if any, one by one, followed by a comparison of their respective means to accelerate propagation.

2.1.1 Scanning

Among those target discovery techniques given above, scanning, which 'entails probing a set of addresses to identify vulnerable hosts', is the most widely employed technique by active worms [1]. This target discovery technique could be implemented differently, which leads to several different types of scanning such as uniform scanning, preferential scanning, sequential scanning, routable scanning, selective scanning, or importance scanning. These different types of scanning approaches are detailed as follows.

The uniform scanning approach probes each IP address from within the whole IPv4 address space with equal probability. Therefore, it needs a perfect random number generator to generate target IP addresses at random. This scanning approach has been employed by famous worms such as the Code-RedI v1 and v2 worms [9]. However, the Code-RedI v1 worm's random number generator was initialized with a fixed seed, so that it spread slowly and never compromised many hosts. Unlike the Code-RedI v1 worm, the Code-RedI v2 worm used a random seed in its random number generator, so that each infected computer tried to infect a different list of randomly generated IP addresses [9]. This seemingly minor change had a major impact: more than 359,000 hosts were infected with the Code-RedI v2 worm in just fourteen hours [9].

The preferential scanning approach probes each IP address from within the whole IPv4 address space with different probabilities. If preference is given to local IP addresses, which means closer IP addresses are to be probed with higher probability, it is termed localized scanning. For example, The CodeRedII worm employed the localized scanning approach by probing IP addresses closer to the currently infected host with higher probability [9]. It is important to note that preference could be given to any factors other than closeness of IP addresses to produce preferential scanning based on that particular factor chosen. For example, if preference is given to IP addresses far away, non-localized scanning could be produced.

The sequential scanning approach probes IP addresses sequentially [10]. In selecting the starting IP address of a sequence, it could choose a closer IP address with higher probability than an IP address farther away. For example, the Blaster worm [10] employed the sequential scanning approach, which for the starting IP address of a sequence, chose the first address of the infected host's class C /24 network with a probability of 0.4 and a random IP address from within the entire IPv4 address space with a probability of 0.6.

The routable scanning approach probes each IP address from within the routable address space rather than the whole IPv4 address space [11]. The information provided by Border Gateway Protocol (BGP) routing tables could be used by this scanning approach to determine which IP addresses are routable since BGP routing tables contain all routable IP addresses [11]. Active worms employing this scanning approach need to carry a list of all routable IP addresses with them, which adds a big payload to them. This big payload will reduce the worms' propagation speed.

Active worms implementing the idea of routable scanning could carry a list of /*n* prefixes that contain all routable IP addresses to reduce the big payload mentioned above. For example, the CodeRedII worm has already implemented this idea by eliminating 127.0.0.0/8 (1 /8 prefix for loopback addresses) and 224.0.0.0/4 (equivalent to 16 /8 prefixes for multicast addresses) from its scanning space, and thus its scanning space was reduced to 93.4% of the entire IPv4 address space (239 out of 256 /8 prefixes) at the cost of carrying a very short list of /8 and /4 prefixes.

The selective scanning approach probes each IP address from within the selected address space rather than the whole IPv4 address space [11]. For example, since BGP routing tables provide detailed information about what Autonomous System (AS) owns a specific network prefix, active worms employing this scanning approach could limit their scanning space to that specific network prefix only in order to propagate only within the AS with that specific network prefix.

All these scanning approaches discussed above have not exploited information on the distribution of vulnerable hosts in their respective scanning space. If information on the distribution of vulnerable hosts is exploited to increase the probability of discovering and vulnerable hosts, a new type of scanning approach emerges, which is termed importance scanning [12].

There exist two cases, in which the importance scanning approach is implemented differently. In one case where the distribution of vulnerable hosts is obtainable in advance, such information will be incorporated into the implementation of importance scanning. In the other case where the distribution of vulnerable hosts is not obtainable in advance, self-learning such information will be incorporated into the implementation of importance scanning [13].

2.1.2 Pre-generated Target List (Hit-List)

A pre-generated target list contains IP addresses of vulnerable hosts obtained in advance, and thus termed so. It is also called a hit-list [14]. An incomplete hit-list could be used to increase the number of initially infected hosts. A complete hit-list creates a ‘flash’ worm, capable of infecting all vulnerable hosts extremely rapidly [15]. They are discussed in more details as follows.

Active worms employing the incomplete hit-list approach could greatly reduce the time needed to infect the first certain number of hosts, if the number of vulnerable hosts contained in the hit-list is greater than or equal to that certain number. This is achieved by infecting those vulnerable hosts contained in the hit-list at first in a very short period. The essence of this target discovery technique is to speed the initial infection by increasing hitting probability (the probability of hitting a vulnerable or infected host) to 100% at the very early stage of active worms’ propagation.

Since a complete hit-list contains IP addresses of all vulnerable hosts, the complete hit-list approach could be used to accelerate active worms' propagation from the beginning to the end, during which period hitting probability remains 100%, while the incomplete hit-list approach can only infect part of all vulnerable hosts in a short time period. Sometimes flash worms cannot reach its full propagation speed due to the bandwidth limit. Furthermore, a complete hit-list of vulnerable hosts is not easy, if not impossible, to obtain, and thus its feasibility is greatly discounted. Therefore, the complete hit-list approach is not a feasible target discovery technique for worm authors.

2.1.3 Internally Generated Target Lists

Internally generated target lists are lists found on infected hosts which contain information about other potential vulnerable hosts. These lists could be used by topological worms, which search for local information to find new targets by trying to discover the local communication topology.

The Morris worm in 1988 employed this target discovery technique to propagate. Since the Internet at that time was very sparse, uniform scanning would be ineffective. E-mail worms (although not active worms) have frequently employed this target discovery technique, as they obtain information about new targets from their victim. Active worms attacking a flaw in peer-to-peer applications could easily get lists of peers from their victims and use those peers as the basis of their attacks, which is another example of employing this target discovery technique.

2.2 Mathematical Models of Active Worms

Due to computer worms' similarity to biological viruses in their spreading behaviors, mathematical models developed to model propagation of infectious diseases have been adapted to model propagation of computer worms [16]. In epidemiology area, both deterministic and stochastic models exist for modeling the spreading of infectious diseases [17-20]. In network security area, both deterministic and stochastic models of active worms based on their respective counterpart in epidemiology area have emerged.

In this section, I will introduce several deterministic and stochastic models of active worms developed by other researchers, followed by a comparison of their respective pros and cons, which naturally leads to tradeoffs among them.

2.2.1 Deterministic Models

Deterministic models of active worms could be further divided into two categories: continuous-time and discrete-time. A continuous-time deterministic model is expressed by single differential equation or set of differential equations, while a discrete-time deterministic model is expressed by single difference equation or set of difference equations. Following are three different examples of continuous-time deterministic models of active worms in a homogeneous system. By 'homogeneous' I mean there is no topology constraint, or, in other words, an infected host is able to directly reach and infect an arbitrary susceptible host. Except the internally generated target lists, all other target discovery techniques employed by active worms satisfy this condition.

I. The Classical Simple Epidemic Model

In the classical simple epidemic model [17-20], all hosts stay in one of only two states at any time: 'susceptible' (denoted by '*S*') or 'infectious' (denoted by '*I*'), and thus it is also called the SI model. This model assumes that once a host is infected by a worm, it will stay in 'infectious' state forever. For a finite population of size *N*, it could be defined by the following single differential equation:

$$\frac{dI(t)}{dt} = \beta I(t)[N - I(t)], \qquad (2.1)$$

where $I(t)$ denotes the number of infectious hosts at time t; and $\beta = \eta$ (average worm scanning rate) / Ω (the size of a worm's scanning space) stands for the pairwise rate of infection in epidemiology studies [21]. At beginning ($t = 0$), $I(0)$ hosts are infectious and the other $N - I(0)$ hosts are all susceptible.

Let $i(t)$ stand for the fraction of the population that are infectious at time t, and thus $i(t) = I(t)/N$, which yields $I(t) = Ni(t)$. Substituting $Ni(t)$ for $I(t)$ in equation (2.1) and rearranging it leads to the differential equation below:

$$\frac{di(t)}{dt} = N\beta i(t)[1 - i(t)]. \qquad (2.2)$$

Differential equation (2.2) has following general analytical solution:

$$i(t) = \frac{e^{N\beta(t-T)}}{1 + e^{N\beta(t-T)}}, \qquad (2.3)$$

which is the logistic equation. A particular analytical solution of differential equation (2.2) given its initial condition $i(0) = I(0)/N$ is as follows:

$$i(t) = \frac{I(0)}{I(0) + [N - I(0)]e^{-N\beta t}}. \qquad (2.4)$$

Staniford *et al.* [14] presented a propagation model for the Code-RedI v2 worm, which is essentially the above classical simple epidemic model.

II. The Classical General Epidemic (Kermack-McKendrick) Model

In the classical general epidemic model (the Kermack-McKendrick model) [17-20], all hosts stay in one of only three states at any time: 'susceptible' (denoted by '*S*'), 'infectious' (denoted by '*I*') or 'removed' (denoted by 'R_I' in this thesis), and thus it is also called the SIR_I model in this thesis. This model does not assume that once a host is infected by a worm, it will stay in 'infectious' state forever. It takes into account the removal process of infectious hosts. However, this model does assume that once a host is removed, it will stay in 'removed' state forever. The removed hosts cannot be infected anymore and they do not try to infect others.

For a finite population of size *N*, this model could be defined by the following set of differential equations:

$$\begin{cases} \dfrac{dI(t)}{dt} = \beta I(t)[N - I(t) - R_I(t)] - \dfrac{dR_I(t)}{dt} \\ \dfrac{dR_I(t)}{dt} = \gamma I(t) \end{cases}, \qquad (2.5)$$

where $I(t)$ denotes the number of infectious hosts at time t; $R_I(t)$ denotes the number of removed hosts from previously infectious hosts at time t; β stands for the pairwise rate of infection; and γ stands for the rate of removal of infectious hosts. It is important to note that a removed host at time t is a host that is once infected but has been removed before time t.

III. The Two-Factor Worm Model

In the two-factor worm model [16], all hosts stay in one of only four states at any time: ‘susceptible’ (denoted by ‘*S*’), ‘infectious’ (denoted by ‘*I*’), ‘removed from susceptible’ (denoted by ‘R_S’ in this thesis) or ‘removed from infectious’ (denoted by ‘R_I’ in this thesis), and thus in this thesis I name it the $\mathrm{SIR_SR_I}$ model. The same as the above classical general epidemic model, this model takes into account the removal process of infectious hosts. Furthermore, it also takes into account the removal process of susceptible hosts, which is one of the two extra factors accounted for in this model. This model assumes that once a host is removed, it will stay in ‘removed’ state forever, whether it has previously been susceptible or infectious. The other of the two extra factors accounted for in this model is the inconstant pairwise rate of infection, which is modeled as a function of time. By considering the above two extra factors, this model is called the two-factor worm model.

The set of differential equations defining the two-factor worm model for a finite population of size *N* is presented as follows:

$$\begin{cases} \dfrac{dI(t)}{dt} = \beta(t)I(t)[N - I(t) - R_S(t) - R_I(t)] - \dfrac{dR_I(t)}{dt} \\ \dfrac{dR_I(t)}{dt} = \gamma I(t) \end{cases}, \qquad (2.6)$$

where $I(t)$ denotes the number of infectious hosts at time t; $R_S(t)$ denotes the number of removed hosts from previously susceptible hosts at time t; $R_I(t)$ denotes the number of removed hosts from previously infectious hosts at time t; $\beta(t)$ stands for the pairwise rate of infection at time t; and γ stands for the rate of removal of infectious hosts.

IV. The Analytical Active Worm Propagation (AAWP) Model

Following is an example of discrete-time deterministic models of active worms in a homogeneous system. It is called the Analytical Active Worm Propagation (AAWP) model [22]. This model applies only to active worms employing the uniform scanning approach since it was derived based on that scanning approach.

Let m_i and n_i denote the total number of vulnerable hosts (including the infected ones) and the number of infected hosts at time tick i ($i \geq 0$), respectively. At beginning when $i = 0$, N hosts are vulnerable or infected ($m_0 = N$) and h hosts are infected ($n_0 = h$). If there are m_i vulnerable hosts (including the infected ones), and n_i infected hosts at time tick i ($i \geq 0$), then on average, at the next time tick, there will be

$$(m_i - n_i)[1-(1-\frac{1}{2^{32}})^{sn_i}] \qquad (2.7)$$

newly infected hosts, where s is the scanning rate [22]. Given death rate d and patching rate p, at the next time tick, there will be $(d + p)\, n_i$ infected hosts that will change to either vulnerable hosts without being infected or invulnerable hosts, and the total number of vulnerable hosts (including the infected ones) will be reduced to $(1 - p)\, m_i$. Therefore, at the next time tick, the number of infected hosts will be

$$n_{i+1} = n_i + (m_i - n_i)[1-(1-\frac{1}{2^{32}})^{sn_i}] - (d+p)n_i . \qquad (2.8)$$

The total number of vulnerable hosts (including the infected ones) at that time tick will be $m_{i+1} = (1 - p)\, m_i$, which yields

$$m_i = (1 - p)^i\, m_0 = (1 - p)^i\, N . \qquad (2.9)$$

Therefore,

$$n_{i+1} = (1 - d - p)n_i + [(1-p)^i N - n_i][1 - (1 - \frac{1}{2^{32}})^{sn_i}], \qquad (2.10)$$

where $i \geq 0$ and $n_0 = h$. Using the set of difference equations, I can find propagation characteristics of active worms employing the uniform scanning approach.

2.2.2 Stochastic Models

Stochastic models of active worms are based on the theory of stochastic processes. All of them are discrete-time in nature. Following are two examples of stochastic models of active worms.

I. The Density-Dependent Markov Jump Process Model

A stochastic density-dependent Markov jump process propagation model [23] of active worms employing the uniform scanning approach drawn from the field of epidemiology [18, 21] is as follows.

Assume N hosts could potentially become infected by the worm. At a given time $t \geq 0$, the set of N potential hosts is split into infected and susceptible subpopulations, represented by $I(t)$ and $S(t)$ respectively. $I(t)$ is the number of hosts which are infected by the worm at time t, and $S(t)$ is the number of hosts which could become infected, but are not at time t. The pair $(S(t), I(t)) = (s, i)$ is the ‘state’ of the epidemic. At time $t = 0$, $(S(0), I(0)) = (s_0, i_0)$, where $i_0 \geq 1$, is the initial state of the epidemic. Due to the random scanning propagation behavior of active worms employing the uniform scanning approach, at time $t > 0$, $S(t)$ and $I(t)$ are random variables. Since infectious hosts are not removed, for all $t \geq 0$, $I(t) + S(t) = N$ holds. If the propagation process is at a state (s, i), the next state must be (s-1, i+1) and the next state after that must be (s-2, i+2) and so on until state (0, N) is reached. From state (0, N) no other state can be reached, so state (0, N) is an absorbing state and almost surely a time t_{fin} is eventually reached such that $(S(t_{fin}), I(t_{fin})) = (0, N)$ [24]. Because the destinations of the infectious packets are selected by the infectious hosts with a uniform distribution, any one infectious packet sent at time t has a probability of $S(t)/2^{32}$ of being sent to a susceptible host. At time t, the $I(t)$ infectious hosts transmit infectious packets each at the rate η, so $(\eta/2^{32})S(t)I(t)$ is the rate at which $(S(t), I(t)) = (s, i)$ goes to (s-1, i+1). From [21], this process can then be modeled as a jump process with a jump intensity:

$$q_{(s_a, i_a)(s_b, i_b)} = \frac{\eta}{2^{32}} s_a i_a , \qquad (2.11)$$

if $s_b = s_a - 1$ and $i_b = i_a + 1$, or 0 otherwise.

The above process is Markovian because at state $(S(t), I(t)) = (s, i)$, the current jump intensity depends only on the current state (s, i) and is independent of the previous states of the process. Consequently, this stochastic epidemic propagation process is by definition a density-dependent Markov jump process because the jump intensity at state (s, i) depends on the ‘densities’ of the number of susceptible hosts s and the number of infected hosts i.

II. The Galton-Watson Markov Branching Process Model

Sellke *et al.* [25] presented a stochastic Galton-Watson Markov branching process model to characterize the propagation of active worms employing the uniform scanning approach. This model is detailed as follows.

Assume N hosts could potentially become infected by the worm. All infected hosts can be classified into generations in the following manner. The initially infected hosts belong to the 0-th generation. All hosts that are directly infected by the initially infected hosts are the 1st generation hosts, regardless of when they are infected. In general, an infected host H_b is an $(n+1)$-th generation host if it is infected directly by a host H_a from the n-th generation. H_b is also called an offspring of H_a. Let ξ be the random variable representing the number of offspring of one infected host scanning M times. During the early phase of the propagation, the vulnerability density d, defined as the probability of successfully finding a vulnerable host in one scan, remains constant ($N/2^{32}$) since the number of infected hosts is much smaller than the number of vulnerable hosts in the population. Thus, during the initial phase of the worm propagation, ξ is a binomial (M, p) random variable. Hence, $P\{\xi = k\} = \binom{M}{k} p^k (1-p)^{M-k}$, $k = 0, 1, \ldots, M$. Let I_n be the number of infected hosts in the n-th generation. I_0 is the number of initially infected hosts. During the early phase of the worm propagation, each infected host in the n-th generation infects a random number of vulnerable hosts, independent of one another, according to the same probability distribution. These newly infected hosts are the (n+1)-th generation hosts. Let $\xi_k^{(n)}$ denote the number of hosts infected by the k-th infected host in the n-th generation. The number of infected hosts in the $(n+1)$-th generation can be expressed as

$$I_{n+1} = \sum_{k=1}^{I_n} \xi_k^{(n)}, \tag{2.12}$$

where $\xi_k^{(n)}$ are independent binomial (M, p) random variables.

The Galton-Watson branching process is a Markov process that models a population in which each individual in generation *n* independently produces some random number of individuals in generation *n*+1, according to a fixed probability distribution that does not vary from individual to individual [26, 27]. During the initial phase of the worm epidemic, each infected host in generation *n* independently produces some random number of infected hosts in generation *n+1*, according to the same probability distribution. Therefore, the early phase of active worms' propagation could be modeled by a stochastic Galton-Watson branching process.

2.2.3 A Comparison of the Mathematical Models

The deterministic models (both continuous-time and discrete-time) of active worms are deterministic abstraction and approximation of a process that is inherently stochastic. Therefore, the stochastic models more accurately describe the propagation dynamics of active worms.

Since propagation of active worms is a discrete event process, the discrete-time deterministic model of active worms (the AAWP model) is more accurate than its continuous-time counterparts (*i.e.*, the SI model, the SIR_I model, and the SIR_SR_I model) in the deterministic regime.

The classical general epidemic model (the SIR_I model) improves the classical simple epidemic model (the SI model) by considering the removal of infectious hosts due to patching. However, this model is still not suitable for modeling propagation of active worms because patching will remove both susceptible hosts and infectious hosts, and the pairwise rate of infection is not constant for rampantly spreading active worms such as the Code-RedI v2 worm and the Slammer worm. The two-factor worm model (the SIR_SR_I model) extends the SIR_I model to account for the removal of susceptible hosts due to patching and at the same time considers the pairwise rate of infection as a function of time rather than a constant. Accounting for these two factors makes this model more accurately reflect the propagation dynamics of the Code-RedI v2 worm [16].

The AAWP model is based on discrete-time and thus more accurate if macro-scope modeling is needed. In this model, a host cannot infect other hosts before it is infected completely. But in models based on continuous-time, a host begins devoting itself to infecting other hosts even though only a 'small part' of it is infected. All of the three continuous-time models do not consider the time an infectious host takes to infect other hosts, while the AAWP model does. The time to infect a host is an important factor for the spread of active worms [28]. Beside, in the AAWP model, the case that worms infect the same destination at the same time is considered, while all of the three continuous-time models ignore this case. In fact, it is not uncommon for a susceptible host to be hit by two or more scans at the same time. Finally, rebooting to change an infectious host's state to 'susceptible' has been taken into account in the AAWP model.

The stochastic density-dependent Markov jump process propagation model could be used to characterize the whole process of active worms' propagation, while the Galton-Watson Markov branching process model presented above only applies to the initial (or early) phase of active worms' propagation due to the assumption that vulnerability density remains constant.

Deterministic approximation is accurate for macro-scale systems. When modeling the propagation of active worms, macro-scale networks are often considered. Therefore, deterministic models of active worms are still applicable to the propagation of active worms spreading in macro-scale networks. The possible variability in the stochastic propagation of active worms is minor under certain conditions, and thus use of deterministic models as a reasonable approximation of the stochastic density-dependent Markov jump process model is justified if those conditions are satisfied [23]. However, in the early phase of the propagation of active worms, the number of infected hosts is very small and thus variance cannot be ignored. Therefore, the stochastic Galton-Watson Markov branching process model, which takes into account the variance, is more accurate than its deterministic counterparts during this period [25].

2.3 Peer-to-Peer Networks and Worms

A peer-to-peer (P2P) network is a type of decentralized and distributed network architecture in which individual nodes in the network (called "peers") act as both suppliers and consumers of resources, in contrast to the centralized client–server model where client nodes request access to resources provided by central servers.

In a peer-to-peer network, tasks (such as searching for files or streaming audio/video) are shared amongst multiple interconnected peers who each makes a portion of their resources (such as processing power, disk storage or network bandwidth) directly available to other network participants, without the need for centralized coordination by servers [29].

A peer-to-peer network is designed around the notion of equal peer nodes simultaneously functioning as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client–server model where communication is usually to and from a central server. A typical example of a file transfer that uses the client-server model is the File Transfer Protocol (FTP) service in which the client and server programs are distinct: the clients initiate the transfer, and the servers satisfy these requests.

Peer-to-peer networks generally implement some form of virtual overlay network on top of the physical network topology, where the nodes in the overlay form a subset of the nodes in the physical network. Data is still exchanged directly over the underlying TCP/IP network, but at the application layer peers are able to communicate with each other directly, via the logical overlay links (each of which corresponds to a path through the underlying physical network). Overlays are used for indexing and peer discovery, and make the P2P system independent from the physical network topology. Based on how the nodes are linked to each other within the overlay network, and how resources are indexed and located, we can classify networks as unstructured or structured (or as a hybrid between the two) [30-32].

Unstructured peer-to-peer networks do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly form connections to each other [33]. Gnutella, Gossip, and Kazaa are examples of unstructured P2P protocols [34].

If a peer wants to find a desired piece of data in the network, the query has to be flooded through the network to find as many peers as possible that share the data. Flooding causes a very high amount of signaling traffic in the network, and does not ensure that search queries will always be resolved. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful. Since there is no correlation between a peer and the content managed by it, there is no guarantee that flooding will find a peer that has the desired data [35]. On the other hand, because there is no structure globally imposed upon them, unstructured networks are easy to build and allow for localized optimizations to different regions of the overlay [36]. Also, because the role of all peers in the network is the same, unstructured networks are highly robust in the face of high rates of "churn" -- that is, when large numbers of peers are frequently joining and leaving the network [37, 38].

In structured peer-to-peer networks the overlay is organized into a specific topology, and the protocol ensures that any node can efficiently search the network for a file/resource, even if the resource is extremely rare [39].

The most common type of structured P2P networks implement a distributed hash table (DHT) [40, 41], in which a variant of consistent hashing is used to assign ownership of each file to a particular peer [42, 43]. This enable peers to search for resources on the network using a hash table: that is, (key, value) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given key [44, 45].

However, in order to route traffic efficiently through the network, nodes in a structured overlay must maintain lists of neighbors that satisfy specific criteria. This makes them less robust in networks with a high rate of churn (i.e. with large numbers of nodes frequently joining and leaving the network) [46]. More recent evaluation of P2P resource discovery solutions under real workloads has pointed out several issues in DHT-based solutions such as high cost of advertising/discovering resources and static and dynamic load imbalance [47].

Notable distributed networks that use DHTs include BitTorrent's distributed tracker, the Kad network, the Storm botnet, YaCy, and the Coral Content Distribution Network. Some prominent research projects include the Chord project, Kademlia, PAST storage utility, P-Grid, a self-organized and emerging overlay network, and CoopNet content distribution system. DHT-based networks have also been widely utilized for accomplishing efficient resource discovery for grid computing systems [48, 49], as it aids in resource management and scheduling of applications.

Harmful data can also be distributed on P2P networks by modifying files that are already being distributed on the network. This type of security breach is created by the fact that users are connecting to untrusted sources, as opposed to a maintained server. In the past this has happened to the FastTrack network when the RIAA managed to introduce faked chunks into downloads and downloaded files (mostly MP3 files). Files infected with the RIAA virus were unusable afterwards or even contained malicious code. The RIAA is also known to have uploaded fake music and movies to P2P networks in order to deter illegal file sharing [50]. Consequently, the P2P networks of today have seen an enormous increase of their security and file verification mechanisms. Modern hashing, chunk verification and different encryption methods have made most networks resistant to almost any type of attack, even when major parts of the respective network have been replaced by faked or nonfunctional hosts.

Some researchers have explored the benefits of enabling virtual communities to self-organize and introduce incentives for resource sharing and cooperation, arguing that the social aspect missing from today's P2P systems should be seen both as a goal and a means for self-organized virtual communities to be built and fostered [51]. Ongoing research efforts for designing effective incentive mechanisms in P2P systems based on principles from game theory are beginning to take on a more psychological and information-processing direction.

Therefore, due to the continuously increasing popularity of P2P networks, P2P worms, which are computer worms spreading themselves by utilizing the features of P2P networks, have been one of the most serious threats to current internet infrastructure.

The common limitation of all of the existing mathematical models of computer worms is that all of them are not applicable to computer worms employing topological scanning. P2P worms are much more difficult to model than non-P2P worms due to their different propagation mechanism from non-P2P worms. All existing mathematical models of computer worms are derived from propagation mechanism of non-P2P worms, and thus they are obviously not applicable to P2P worms since they do not embody propagation mechanism of P2P worms.

The novel logic matrix approach proposed in this thesis models the propagation processes of P2P worms by difference equations of logic matrix, which are essentially discrete-time deterministic propagation models of P2P worms. The proposed models are suitable for modeling P2P worms because these models take into account topology of a P2P network.

I use an experimental approach to conducting the research. According to [10-13, 16, 22, 55, 56], similar approaches have been effectively applied in their respective study carried out in this domain.

Chapter 3 Propagation Acceleration by Employing Multiple Techniques

According to Xiang *et al.*[52], an active worm is not limited to employing single target discovery technique only, and thus future active worms could employ multiple target discovery techniques simultaneously in an attempt to accelerate their propagation [53]. To find an effective countermeasure to this sort of future worms, we must study their propagation mechanisms thoroughly, and investigate their propagation characteristics under various scenarios. Therefore, I studied propagation mechanisms of active worms employing a combination of two or three different target discovery techniques from attackers' perspective. I also performed a series of simulation experiments to investigate their propagation characteristics under various scenarios.

3.1 Theoretical Studies

3.1.1 The Significance of Target Discovery Techniques

The life cycle of a worm from when it is released to when it finishes infecting vulnerable hosts, consists of the initialization phase, the network propagation phase, and the payload activation phase [54]. Once a host is infected, each worm instance begins with the initialization phase. Following the initialization phase is the network propagation phase, which is the phase that encompasses the behavior that describes how a worm moves through a network. In this phase, a worm attempts to infect its target hosts by performing a sequence of actions including target acquisition, network reconnaissance, attack, and infection. Once a target host is infected, the initialization phase of the new instance of the worm begins. Any time following the initialization phase comes the payload activation phase. To date, payloads that significantly affect propagation characteristics of a worm have been rare. The Code Red worms, the Slammer worm, and the Witty worm are all examples of payloads that occurred to the exclusion of network propagation. Figure 3.1 illustrates the relationship among the phases in the life cycle of worms and the actions performed in the network propagation phase.

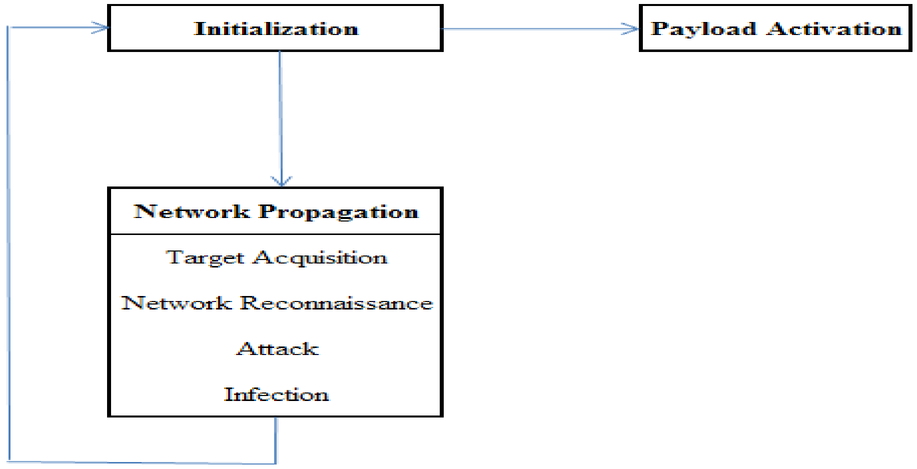


Figure 3.1: Phases in the life cycle of worms and actions in the network propagation phase

Since target acquisition and network reconnaissance together essentially dictate target discovery technique(s) employed by a worm, the significance of target discovery techniques in shaping a worm's propagation characteristics was derived from the life cycle of worms in [52].

This chapter explores how to accelerate the propagation of active worms by employing multiple target discovery techniques.

3.1.2 The Significance of Shortening the Slow Start Phase

For the classical simple epidemic model given in the previous chapter, Figure 3.2 shows the dynamics of I_t -- denoted by $I(t)$ in equation (2.1) -- as time goes on for a certain set of parameters [55, 56].

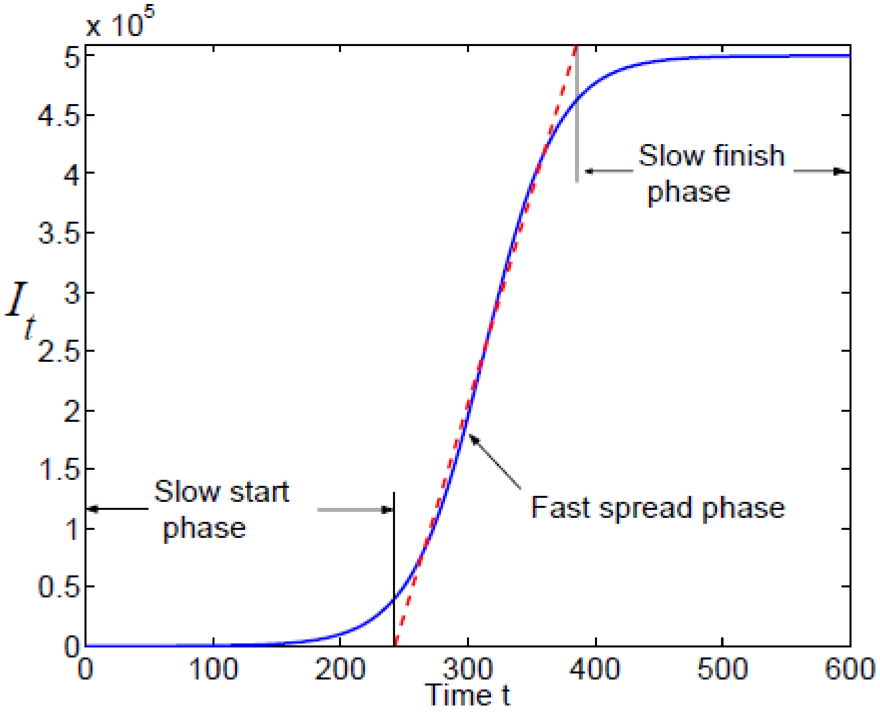


Figure 3.2: Propagation curve of the classical simple epidemic model

According to Figure 3.2, we can roughly partition a worm's propagation into three phases: the slow start phase, the fast spread phase, and the slow finish phase. During the slow start phase, since $I_t << N$, model (2.1) becomes

$$\frac{dI_t}{dt} \approx N\beta I_t, \tag{3.1}$$

which means that the number of infectious hosts increases exponentially approximately. After a certain number of susceptible hosts are infected and then participate in infecting others, the worm enters its fast spread phase where susceptible hosts are infected at a fast, nearly constant rate. When most susceptible hosts have been infected, the worm enters its slow finish phase because the few susceptible hosts leftover are difficult for the worm to find.

Since $i_t = \frac{I_t}{N}$, the shape of i_t -- denoted by $i(t)$ in equations (2.2), (2.3) and (2.4) -- against t will be exactly the same as that of I_t with scale-down by a factor of N. In other words, i_t is $\frac{I_0}{N}$ at $t = 0$ and converges to 1 when t goes to positive infinite.

According to equation (2.4),

$$i_t = \frac{I_0}{I_0 + (N - I_0)e^{-N\beta t}}. \tag{3.2}$$

Letting $a = I_0$, $b = N - I_0$, and $c = -N\beta$ will transform equation (3.2) to

$$i_t = \frac{a}{a + be^{ct}}. \tag{3.3}$$

The first derivative of i_t is worked out and shown as follows:

$$\frac{di_t}{dt} = \frac{-abce^{ct}}{(a + be^{ct})^2}. \tag{3.4}$$

We can then work out the second derivative of i_t and let it equal to 0:

$$\frac{d^2 i_t}{dt^2} = 0. \tag{3.5}$$

This will lead to $\frac{di_t}{dt} = -\frac{c}{4}$ and $i_t = 50\%$. In other words, the maximum rate at which susceptible hosts are infected equals to $-\frac{c}{4} = \frac{N\beta}{4}$, and this maximum rate is achieved at the moment when 50% of susceptible hosts are infected.

I define fast spread as that with a rate not less than 50% of the maximum rate, which is $-\frac{c}{8}$. $\frac{di_t}{dt} = \frac{-abce^{ct}}{(a+be^{ct})^2} = -\frac{c}{8}$ leads to $i_t \approx 15\%, 85\%$. In other words, according to our definition of fast spread, when less than 15% of susceptible hosts are infected, the worm is in its slow start phase; when more than 85% of susceptible hosts are infected, the worm is in its slow finish phase; in between, the worm is in its fast spread phase.

It is obvious that in order to accelerate a worm's propagation, we must try to let the worm infect the first 15% susceptible hosts and enter its fast spread phase as soon as possible. On the other hand, the last 15% susceptible hosts leftover are not important for attackers if infection of 85% susceptible hosts will serve their purposes, which is usually the case.

Next chapter explores how to accelerate the propagation of active worms by shortening the slow start phase.

3.1.3 The Proposed Propagation Model of Active Worms

Here, I present the proposed discrete-time deterministic Compensation Factor Adjusted Propagation (CFAP) model of active worms employing uniform scanning as their target discovery technique. Let η and Ω stand for an active worm's scanning rate and scanning space, respectively. For a finite population of size *N*, assume at time *t*, there exist I_t infectious hosts. Then at time $t+\frac{1}{\eta}$, by assuming the probability of different infectious hosts hitting the same susceptible host to be 0, there will be $I_{t+1} = I_t + \Delta I_t$ infectious hosts, where

$$\Delta I_t = \frac{I_t(N - I_t)}{\Omega}. \tag{3.6}$$

During the process that more and more susceptible hosts are infected and then participate in infecting others, the probability of different infectious hosts hitting the same susceptible host is not a constant. Therefore, the actual number of newly infected hosts is less than that predicted by equation (3.6). Here, I introduce a compensation factor denoted by C_t to account for the difference between them, which varies as time goes on. Therefore, the discrete-time deterministic CFAP model could be described by the following difference equation:

$$I_{t+1} = I_t + \frac{I_t(N - I_t)}{\Omega} - C_t. \tag{3.7}$$

There exist two methods to determine C_t , which are mathematical analysis or simulation. To predict C_t in a closed form (*i.e.*, with no or very little iteration), mathematical analysis is usually employed. However, in some situations it could be very difficult, if not impossible, to derive a formula of C_t as a function of t. Then, I have to perform simulation experiments to find approximate value of C_t at each time t.

3.2 Simulation Experiments

3.2.1 Effective Tools to Understand Complex Processes

There are four different ways to study the characteristics of a piece of self-propagating code, which are using test beds, performing real world experiments, creating mathematical models, and performing simulation experiments [57].

Test beds allow to actually set free a piece of self-replicating code in an isolated and limited environment and to observe its behavior. The most obvious limit of test beds is that they cannot be created in a size approaching the size of the Internet. Using the Internet itself to do real world experiments is not an option for scientific study because of the damage being done. The most powerful approach is probably the creation of realistic mathematical models that allow behavior prediction in a closed form. The problem with this approach is that such models are not generally available and are usually hard or even impossible to create.

In a sense simulation is a mathematical model in which some of the functions used rely heavily on iteration. In order to reduce computational complexity, abstraction and approximation of the inner mechanisms of the object studied is often used. This allows computation of functions that are not well understood in a mathematical sense. Here, the analytical approach of mathematical modeling is replaced with an experimental approach, in which scenarios are simulated and then analyzed. Simulation experiments are often very effective tools to understand complex processes.

3.2.2 Efficient Way to Understand Complex Processes

I systematically examined propagation characteristics of active worms employing the single target discovery technique only, and a combination of two or three different target discovery techniques by conducting a series of simulation experiments under various scenarios. In order to reduce simulation time, I performed our simulation experiments in a class A /8 subnet. In other words, I used scale-down by a factor of $\frac{1}{2^8}$ to explore worm dynamics. According to Weaver *et al.* [58], scale-down introduces two notable artifacts: a bias towards more rapid propagation (propagation curve being shifted to the left due to scale-up of the density of initially infected hosts), and an increase in stochastic effects. Although these artifacts are significant, scale-down can still capture general behavior as long as the scale-down factor is not too extreme [58]. Therefore, scale-down is an efficient way to understand complex processes if the scale-down factor is appropriately chosen.

The simulation experiments were based on the assumption that susceptible hosts are uniformly distributed in the above address space with vulnerability density approximately equivalent to that of the Slammer worm. I also assumed average worm scanning rate to be equivalent to the Slammer's as well. All simulations started with only 1 initially infected host, which is equivalent to 2^8 initially infected hosts in the Slammer's case. Outputs from all simulations are the numbers of infected hosts against time.

These assumptions are necessary to make the simulation experiments not overwhelmingly complicated. Since simulation experiments are based on similar assumptions, simulation results are comparable. Besides, these assumptions capture the major features of computer worms while ignore the minor ones, so they are realistic and will not make the simulation results much different from real cases.

In order to eliminate variation in results from different simulation runs for each certain scenario, I performed 10 simulation runs for each scenario using the simulator implemented in C programming language custom made for the simulation experiments. Results from all simulation runs are then averaged to produce final result for each scenario. I repeated the simulation experiments 10 times and found that the standard deviation of the average to be 0, which indicated that stochastic effects could be eliminated, and the scale-down factor chosen was appropriate.

3.2.3 Scenarios Simulated

I. Simple Scenarios

Before I studied propagation characteristics of active worms employing a combination of two or three different target discovery techniques, I had studied propagation characteristics of active worms employing only one of the following target discovery techniques: uniform scanning; a complete hit-list; or internally generated target lists.

The above three kinds of active worms became the first 3 scenarios to be simulated, which are summarized in Table 3.1. Propagation rate of active worms employing uniform scanning only was the baseline to be compared to. Since an incomplete hit-list when not combined with any other target discovery technique(s) cannot let a worm infect more hosts than those in the list, in practice it must be combined with other target discovery technique(s). Therefore, I chose a complete hit-list as one of the above 3 fundamental target discovery techniques. Size of internally generated target list might vary with different hosts. For simplicity, average size of internally generated target lists was chosen as a candidate parameter, whose influence on a worm's propagation characteristics was to be investigated.

Table 3.1: A summary of the 3 simple scenarios simulated

Scenario Code	**Target Discovery Technique Employed**
U	Uniform Scanning Only
H100%	A Complete Hit-list Only
I1	Internally Generated Target Lists Only with Average Size of 1
I2	Internally Generated Target Lists Only with Average Size of 2
I3	Internally Generated Target Lists Only with Average Size of 3

Table 3.2: A summary of simulation results of the 3 simple scenarios

Scenario Code	**Average Time (in seconds) to Infect 99% Susceptible Hosts**
U	142
H100%	1
I1	Indefinite (maximum infection rate of 7% achieved in 1 second)
I2	Indefinite (maximum infection rate of 79% achieved in 1 second)
I3	Indefinite (maximum infection rate of 94% achieved in 1 second)

According to the results (Table 3.2) from our simulation experiments, a complete hit-list makes a worm propagate extremely rapidly. However, the feasibility of this approach is discounted by the extreme difficulties that will be encountered by attackers in gathering such a list. Due to their exactly same propagation mechanism, an incomplete hit-list lets a worm infect all susceptible hosts in the list as soon as a complete hit-list does. Therefore, an incomplete hit-list is a more feasible approach. It is obvious that active worms only employing internally generated target lists with average size not greater than 3 cannot achieve infection of over 99% susceptible hosts. An explanation to this phenomenon could be that less than 99% of all susceptible hosts are in the combined internally generated target lists of all susceptible hosts infected. However, average size of internally generated target lists has a great influence on the maximum infection rate (maximum percentage of susceptible hosts a worm can infect). A slight increase in average size from 1 to 3 leads to a dramatic increase in the maximum infection rate. Furthermore, maximum infection rates are achieved in 1 second for all average sizes (1, 2, or 3). As I mentioned earlier in this paper, infection of 85% susceptible hosts would usually serve attackers' purposes. Therefore, internally generated target lists with average size of 3 (with maximum infection rate of 94%) could be employed by active worms to accelerate their propagation. A comparison of propagation curves of the 3 simple scenarios is illustrated by Figure 3.3.

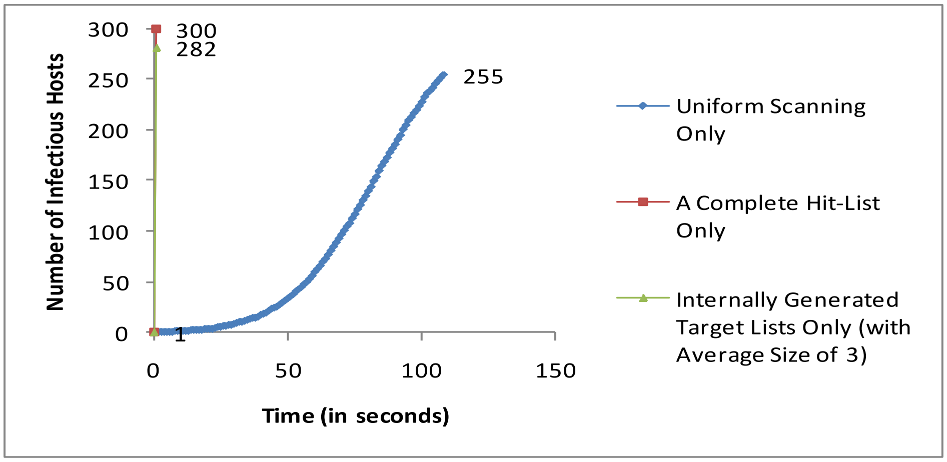


Figure 3.3: A comparison of propagation curves of the 3 simple scenarios

II. Scenarios with Moderate Complexity

Then, propagation characteristics of active worms employing a combination of two different target discovery techniques formed the focus of this research. As I mentioned earlier in this chapter, in order to accelerate a worm's propagation, we must try to let the worm infect the first 15% susceptible hosts and enter its fast spread phase as soon as possible. According to the simulation results of the above 3 simple scenarios, both an incomplete hit-list and internally generated target lists can let a worm infect a certain percentage of susceptible hosts in just one second. Therefore, each of these two target discovery techniques could be followed by uniform scanning to let the worm infect those susceptible hosts leftover. In our simulation experiments, active worms employing an incomplete hit-list followed by uniform scanning as their target discovery techniques would sequentially probe all those hosts in the hit-list prior to employing uniform scanning. Active worms employing internally generated target lists followed by uniform scanning would sequentially probe all those hosts in the target lists generated in process prior to employing uniform scanning.

The above two kinds of active worms formed the basis of the 6 scenarios with moderate complexity to be simulated, which are summarized in Table 3.3. Since I intended to shorten a worm's slow start phase, in which less than 15% of susceptible hosts are infected, an incomplete hit-list with size up to 15% of the number of all susceptible hosts was employed. Both size of incomplete hit-list and average size of internally generated target lists were candidate factors whose influences on a worm's propagation characteristics were to be investigated. I have simulated a limited number of scenarios. More scenarios could be investigated to determine the relationship between average time to infect 99% susceptible hosts and size of hit-list, and the relationship between average time to infect 99% susceptible hosts and average size of internally generated target lists.

Table 3.3: A summary of the 6 simulated scenarios with moderate complexity

Scenario Code	Target Discovery Techniques Employed
H5%+U	An Incomplete Hit-list with Size = 5% of the Number of All Susceptible Hosts; Followed by Uniform Scanning
H10%+U	An Incomplete Hit-list with Size = 10% of the Number of All Susceptible Hosts; Followed by Uniform Scanning
H15%+U	An Incomplete Hit-list with Size = 15% of the Number of All Susceptible Hosts; Followed by Uniform Scanning
I1+U	Internally Generated Target Lists with Average Size of 1;Followed by Uniform Scanning
I2+U	Internally Generated Target Lists with Average Size of 2; Followed by Uniform Scanning
I3+U	Internally Generated Target Lists with Average Size of 3; Followed by Uniform Scanning

Table 3.4: A summary of simulation results of the 6 scenarios with moderate complexity

Scenario Code	Average Time (in seconds) to Infect 99% Susceptible Hosts
H5%+U	99
H10%+U	89
H15%+U	85
I1+U	60
I2+U	36
I3+U	21

According to the results (Table 3.4) from the simulation experiments, an incomplete hit-list with size of 5% of the number of all susceptible hosts followed by uniform scanning accelerates a worm’s propagation dramatically. However, this approach’s capability to accelerate active worms’ propagation is diminishing while size of the hit-list is increasing. Active worms employing internally generated target lists followed by uniform scanning performed especially well under all average sizes (1, 2, or 3) of the target lists. Here, average size of the target lists has a great influence on a worm’s propagation rate. The larger the average size becomes, the faster the worm propagates.

I have also investigated propagation characteristics of active worms employing both an incomplete hit-list and internally generated target lists as their target discovery techniques. According to our simulation results of the 3 simple scenarios, an incomplete hit-list ought to be employed prior to internally generated target lists because generally the former is more effective to boost the number of initially infected hosts. Therefore, in our simulation experiments, active worms employing both an incomplete hit-list and internally generated target lists as their target discovery techniques would sequentially probe all those hosts in the hit-list prior to sequentially probing all those hosts in the target lists generated in process. Our simulation results show that active worms employing internally generated target lists with average size not greater than 3 cannot achieve infection of over 99% susceptible hosts, even if the number of initially infected hosts is boosted by an incomplete hit-list of size up to 15% of the number of all susceptible hosts. A simple and efficient way to infect those leftover susceptible hosts is by uniform scanning. Therefore, I believe uniform scanning is an indispensable elementary target discovery technique of active worms.

III. Complex Scenarios

Finally, propagation characteristics of active worms employing a combination of three different target discovery techniques were examined. In our simulation experiments, active worms employing an incomplete hit-list followed by internally generated target lists followed by uniform scanning as their target discovery techniques would sequentially probe all those hosts in the hit-list prior to prior to sequentially probing all those hosts in the target lists generated in process. Once those lists were exhausted, they would start uniform scanning.

The above kind of active worm formed the basis of the 9 complex scenarios to be simulated, which are summarized in Table 3.5. Both size of incomplete hit-list and average size of internally generated target lists were candidate factors whose influences on a worm's propagation characteristics were to be investigated. I have simulated a limited number of scenarios. More scenarios could be investigated to determine the relationship between average time to infect 99% susceptible hosts and size of hit-list and average size of internally generated target lists.

Table 3.5: A summary of the 9 complex scenarios simulated

Scenario Code	Target Discovery Technique(s) Employed
H5%+I1+U H5%+I2+U H5%+I3+U	An Incomplete Hit-list with Size = 5% of the Number of All Susceptible Hosts; Followed by Internally Generated Target Lists with Average Size of 1, 2, or 3; Followed by Uniform Scanning
H10%+I1+U H10%+I2+U H10%+I3+U	An Incomplete Hit-list with Size = 10% of the Number of All Susceptible Hosts; Followed by Internally Generated Target Lists with Average Size of 1, 2, or 3; Followed by Uniform Scanning
H15%+I1+U H15%+I2+U H15%+I3+U	An Incomplete Hit-list with Size = 15% of the Number of All Susceptible Hosts; Followed by Internally Generated Target Lists with Average Size of 1, 2, or 3; Followed by Uniform Scanning

Table 3.6: A summary of simulation results of the 9 complex scenarios

Scenario Code	Average Time (in seconds) to Infect 99% Susceptible Hosts
H5%+I1+U	54
H5%+I2+U	34
H5%+I3+U	18
H10%+I1+U	55
H10%+I2+U	36
H10%+I3+U	19
H15%+I1+U	53
H15%+I2+U	35
H15%+I3+U	18

According to the results (Table 3.6) from our simulation experiments, an additional incomplete hit-list only accelerates a worm’s propagation slightly, compared to the results of the last 3 scenarios in Table 3.4. Increasing size of the hit-list has little effect on a worm’s rate of propagation. However, average size of internally generated target lists has a great influence on a worm’s rate of propagation. The larger the average size becomes, the faster the worm propagates. It could also be found that further increasing size of the additional incomplete hit-list might slow down the propagation, which could be explained by overlap of hosts in the hit-list and the internally generated target lists. In other words, the results indicate the combination of the three different target discovery techniques is not the best for attackers taking into account the added effort they have to make to build the worm. I found internally generated target lists with average size of 3 followed by uniform scanning is the most effective and efficient among all approaches examined in this paper to accelerate propagation of active worms.

Chapter 4 Propagation Acceleration by Shortening the Slow Start Phase

Zou *et al.* [55, 56] partition a worm’s propagation into three phases: the slow start phase, the fast spread phase, and the slow finish phase. According to [53], the best way to accelerate a worm’s propagation is to shorten the period of the slow start phase and let the worm enter its fast spread phase as soon as possible. Xiang *et al.* [52] point out that an active worm is not limited to employing single target discovery technique only, and thus future active worms could employ multiple target discovery techniques simultaneously in an attempt to accelerate their propagation. I believe this strategy could also be utilized to shorten the slow start phase in the propagation of active worms [59]. Therefore, I studied propagation mechanisms of active worms employing a combination of two different target discovery techniques in an attempt to shortening its slow start phase. I also performed a series of simulation experiments to investigate their propagation characteristics under various scenarios.

4.1. Preliminary Studies

4.1.1 The Approach Chosen and Set-up of the Simulation Experiments

I performed simulation experiments to understand the propagation characteristics of active worms. In order to reduce simulation time, I performed our simulation experiments in a class A /8 subnet. In other words, I used scale-down by a factor of $1/2^8$ to explore worm dynamics.

Our simulation experiments were based on the same assumptions as those on which the simulation experiments described in Chapter 3 were based. They include assumptions regarding distribution of susceptible hosts, vulnerability density and average worm scanning rate, details of which were given in the setup described in Chapter 3. All simulations started with only 1 initially infected host, which is equivalent to 2^8 initially infected hosts in the Slammer's case. Outputs from all simulations are the numbers of infected hosts against time.

In order to eliminate variation in results from different simulation runs for each certain scenario, I performed 10 simulation runs for each scenario using the simulator implemented in C programming language custom made for our simulation experiments. Results from all simulation runs are then averaged to produce final result for each scenario.

4.1.2 Potential Approaches to Shortening the Slow Start Phase

Before I studied propagation characteristics of active worms employing a combination of two different target discovery techniques in an attempt to shorten it slow start phase, I had studied propagation characteristics of active worms employing only one of the following target discovery techniques: uniform scanning; a complete hit-list; or internally generated target lists.

The above three kinds of active worms formed the basis of the scenarios to be simulated, which are summarized in Table 4.1. Propagation rate of active worms employing uniform scanning only was our baseline to be compared to. Since an incomplete hit-list only cannot let a worm infect more hosts than those in the list, in practice it must be combined with other target discovery technique(s). Therefore, I chose a complete hit-list as one of the above 3 fundamental target discovery techniques. Average size of internally generated target lists was a candidate factor whose influence on a worm's propagation characteristics was to be investigated.

Table 4.1: A list of the scenarios simulated

Scenario	Target Discovery Technique Employed
Uniform	Uniform scanning only
Hit-List_100%	A complete hit-list only
Target_Lists_1	Internally generated target lists (with average size of 1) only
Target_Lists_2	Internally generated target lists (with average size of 2) only
Target_Lists_3	Internally generated target lists (with average size of 3) only
Target_Lists_4	Internally generated target lists (with average size of 4) only
Target_Lists_5	Internally generated target lists (with average size of 5) only
Target_Lists_6	Internally generated target lists (with average size of 6) only

Table 4.2: A summary of simulation results of the scenarios simulated

Scenario	**Average Time (in seconds) to Infect 99% Susceptible Hosts**
Uniform	142
Hit-List_100%	1
Target_Lists_1	Indefinite (average maximum infection rate 7% achieved in 1 second)
Target_Lists_2	Indefinite (average maximum infection rate 79% achieved in 1 second)
Target_Lists_3	Indefinite (average maximum infection rate 94% achieved in 1 second)
Target_Lists_4	1 with probability 0.45 and indefinite with probability 0.55 (average maximum infection rate 98% achieved in 1 second)
Target_Lists_5	1 with probability 0.97 and indefinite with probability 0.03 (average maximum infection rate 99% achieved in 1 second)
Target_Lists_6	1 with probability 1 (average maximum infection rate 100% achieved in 1 second)

According to the results (Table 4.2) from our simulation experiments, a complete hit-list makes a worm propagate extremely rapidly. However, the feasibility of this approach is discounted by the extreme difficulties that will be encountered by attackers in gathering such a list. Due to their exactly same propagation mechanism, an incomplete hit-list lets a worm infect all susceptible hosts in the list as soon as a complete hit-list does. Therefore, an incomplete hit-list is a more feasible approach.

According to results from the simulation experiments, active worms only employing internally generated target lists with average size not greater than 5 cannot achieve infection of over 99% susceptible hosts. An explanation to this phenomenon could be that less than 99% of all susceptible hosts are in the combined internally generated target lists of all susceptible hosts infected. However, average size of internally generated target lists has a great influence on the average maximum infection rate (average maximum percentage of susceptible hosts a worm can infect). A slight increase in the average size from 1 to 6 leads to a dramatic increase in the average maximum infection rate. Furthermore, average maximum infection rates are achieved in 1 second for all average sizes (1 to 6). As I mentioned in Chapter 3, infection of 85% susceptible hosts would usually serve attackers' purposes. Therefore, internally generated target lists with average size of 3 or greater (with average maximum infection rate of 94% or greater) could be employed by active worms to shorten its slow start phase.

We can see from the above simulation results that both a complete hit-list and internally generated target lists with average size of 6 (2% of the number of all susceptible hosts) are able to let the worm infect all susceptible hosts in just 1 second. Compared to uniform scanning only, both of them will reduce a worm's average time to infect over 99% susceptible hosts by 141 seconds. However, the same as a complete hit-list is hard to obtain, target lists with average size of 6 are not always available. Since neither an incomplete hit-list with size less than 99% of the number of all susceptible hosts nor target lists with average size of 5 or less are able to infect over 99% susceptible hosts, both of them have to be combined with other target discovery technique(s) to enable a worm to infect those leftover susceptible hosts. I believe uniform scanning is the simplest and best way to achieve that purpose. Therefore, I studied propagation characteristics of active worms employing an incomplete hit-list with increasing size and internally generated target lists with increasing average size followed by uniform scanning. In the next section, I will detail our investigation of employing the strategy proposed in chapter 3 to shorten the slow start phase in the propagation of active worms.

4.2 Employing Multiple Techniques to Shorten the Slow Start Phase

After having identified several target discovery techniques that could be utilized to shorten the slow start phase in the propagation of active worms, a reasonably comprehensive investigation of employing the strategy proposed in Chapter 3 to shorten the slow start phase is carried out by conducting a series of simulation experiments. I systematically examined propagation characteristics of active worms employing an incomplete hit-list with increasing size followed by uniform scanning and internally generated target lists with increasing average size followed by uniform scanning by conducting a series of simulation experiments under various scenarios.

4.2.1 An Incomplete Hit-List with Increasing Size

As implied in the last section, the time needed to infect the first certain percentage (approximately 15% according to our calculations) of susceptible hosts dominates the time to infect over 85% susceptible hosts. Therefore, I only need to investigate rate of propagation for an incomplete hit-list with size up to 15% of the number of all susceptible hosts.

In our simulation experiments, active worms employing an incomplete hit-list followed by uniform scanning as their target discovery techniques would sequentially probe all those hosts in the hit-list prior to employing uniform scanning. I started with a hit-list with size of 1% of the number of all susceptible hosts and kept increasing its size by 1% at a time until it reached 15%.

Scenarios simulated and their results are summarized in Table 4.3. I also worked out decreases in average time required to infect a certain percentage (say 99%) of susceptible hosts. For each additional 1% increasing of size of the hit-list, decreases in average time required to infect over 99% of susceptible hosts are significantly unequal. Therefore, a cost (increase in hit-list size) and benefit (decrease in time required to infect a certain percentage of susceptible hosts) analysis was conducted to discover the most cost-effective and cost-efficient hit-list size.

Table 4.3: A summary of the scenarios simulated and their results (1)

Scenario	**Average Time (in seconds) to Infect 99% Susceptible Hosts**	**Decrease in Average Time (in seconds) to Infect 99% Susceptible Hosts Caused by Increase by 1% in Hit-List Size**
Hit-List_0%+Uniform	142	
Hit-List_1%+Uniform	120	22
Hit-List_2%+Uniform	110	10
Hit-List_3%+Uniform	110	0
Hit-List_4%+Uniform	108	2
Hit-List_5%+Uniform	100	8
Hit-List_6%+Uniform	96	4
Hit-List_7%+Uniform	95	1
Hit-List_8%+Uniform	93	2
Hit-List_9%+Uniform	94	-1
Hit-List_10%+Uniform	90	4
Hit-List_11%+Uniform	94	-4
Hit-List_12%+Uniform	88	6
Hit-List_13%+Uniform	88	0
Hit-List_14%+Uniform	81	7
Hit-List_15%+Uniform	86	-5
Hit-List_20%+Uniform	80	1
Hit-List_25%+Uniform	80	0
Hit-List_30%+Uniform	72	1
Hit-List_35%+Uniform	73	0
Hit-List_40%+Uniform	70	0
Hit-List_45%+Uniform	63	1
Hit-List_50%+Uniform	61	0

According to the results from our simulation experiments, an incomplete hit-list with size of 1% of the number of all susceptible hosts followed by uniform scanning accelerates a worm's propagation dramatically. This strategy reduces a worm's average time required to infect over 99% susceptible hosts by 22 seconds compared to uniform scanning only. (denoted by Hit-List_0%+Uniform in Table 4.3). However, this approach's capability to accelerate active worms' propagation is diminishing while size of hit-list keeps increasing.

In this chapter, I define a cost-effective action as one with a ratio of additional benefit gained to additional cost paid for that benefit gained not less than one third of that of the best case. The threshold is set arbitrarily. Since simulation results are compared to each other, relative effectiveness rather than absolute effectiveness matters. The set threshold does not affect the relative effectiveness rankings of simulation results. Because my cost and benefit analysis is unique, there are no existing benchmarks in terms of cost-effectiveness. Therefore, I do not compare my simulation results with those in benchmark in terms of cost-effectiveness.

To simplify our cost and benefit analysis, I assume the cost involved in discovering each additional susceptible host is same. Based on this assumption, we will have to make same effort in increasing size of hit-list by each percentage. However, our benefit gained (reduced average time to infect over 99% susceptible hosts) from each percentage increase in hit-list size is not same at all. The maximum benefit (22 seconds) is gained when the first 1% susceptible hosts are found and incorporated into a worm's hit-list. When the fifth 1% susceptible hosts are found and incorporated into a worm's hit-list, we still gain a benefit of 8 seconds, which is not less than one third of 22 seconds. Therefore, it is still cost effective according to the definition given above. However, after that, benefits gained are all less than one third of 22 seconds, which indicates they are all not cost-effective.

Based on the above cost and benefit analysis, I suggest a hit-list with size of 5% of the number of all susceptible hosts followed by uniform scanning is a cost-effective target discovery technique that could be employed by active worms to accelerate their propagation by approximately 30% ($\frac{142-100}{142}$) compared to employing uniform scanning only.

4.2.2 Internally Generated Target Lists with Increasing Average Size

As I discovered in the last section, active worms only employing internally generated target lists with average size not greater than 5 cannot achieve infection of over 99% susceptible hosts, but they can achieve their respective average maximum infection rates in just 1 second for all average sizes (1 to 6). Therefore, internally generated target lists with average size of not greater than 5 could be employed by active worms to shorten their slow start phase dramatically. I believe if this target discovery technique is combined with uniform scanning, it will enable a worm to infect over 99% susceptible hosts.

In our simulation experiments, active worms employing internally generated target lists followed by uniform scanning as their target discovery techniques would sequentially probe all those hosts in the target lists generated in process prior to employing uniform scanning. I started with target lists of average size 1 and kept increasing their average size by 1 at a time until it reached 5. Since internally generated target lists with average size greater than 5 will enable a worm to infect over 99% susceptible hosts in a very short time period, it is not necessary to combine them with uniform scanning.

Scenarios simulated and their results are summarized in Table 4.4. I also worked out decreases in average time required to infect a certain percentage (say 99%) of susceptible hosts. For each additional increase by 1 in average size of internally generated target lists, decreases in average time required to infect over 99% of susceptible hosts are significantly unequal. Therefore, a cost (increase in average size of internally generated target lists) and benefit (decrease in time required to infect a certain percentage of susceptible hosts) analysis was conducted to discover the most cost-effective and cost-efficient average size of internally generated target lists.

Table 4.4: A summary of the scenarios simulated and their results (2)

Scenario	**Average Time (in seconds) to Infect 99% Susceptible Hosts**	**Decrease in Average Time (in seconds) to Infect 99% Susceptible Hosts Caused by Increase by 1 in Average Size of Internally Generated Target Lists**
Target_Lists_0+Uniform	142	
Target_Lists_1+Uniform	60	82
Target_Lists_2+Uniform	36	24
Target_Lists_3+Uniform	21	15
Target_Lists_4+Uniform	9	12
Target_Lists_5+Uniform	2	7

According to the results from our simulation experiments, internally generated target lists with average size of 1 followed by uniform scanning accelerates a worm's propagation dramatically. This strategy reduces a worm's average time required to infect over 99% susceptible hosts by 82 seconds compared to uniform scanning only. (denoted by Target-Lists_0+Uniform in Table 4.4). However, this approach's capability to accelerate active worms' propagation is diminishing while average size of internally generated target lists keeps increasing.

To simplify our cost and benefit analysis, I assume the cost involved in increasing average size of internally generated target lists by 1 remains unchanged. Based on this assumption, we will have to make same effort in increasing average size of internally generated target lists by 1. However, our benefit gained (reduced average time to infect over 99% susceptible hosts) from each increase by 1 of average size of internally generated target lists is not same at all. The maximum benefit (82 seconds) is gained when the average size of internally generated target lists is increase from 0 to 1. When the average size of internally generated target lists is increased from 1 to 2, we gain a benefit of 24 seconds, which is less than one third of 82 seconds. Therefore, it is not cost effective according to our definition given in the last sub-section. After that, benefits gained are all less than one third of 82 seconds, which indicates they are all not cost-effective.

Based on the above cost and benefit analysis, I suggest average size of internally generated target lists of 1 (approximately 0.33% of the number of all susceptible hosts) followed by uniform scanning is a cost-effective target discovery technique that could be employed by active worms to accelerate their propagation by approximately 60% ($\frac{142-60}{142}$) compared to employing uniform scanning only.

4.2.3 A Combination of the Above Two Approaches

I have also investigated propagation characteristics of active worms employing both an incomplete hit-list and internally generated target lists as their target discovery techniques. According to our simulation results, an incomplete hit-list ought to be employed prior to internally generated target lists because generally the former is more effective to boost the number of initially infected hosts. Therefore, in our simulation experiments, active worms employing both an incomplete hit-list and internally generated target lists as their target discovery techniques would sequentially probe all those hosts in the hit-list prior to sequentially probing all those hosts in the target lists generated in process. Our simulation results show that active worms employing internally generated target lists with average size not greater than 3 cannot achieve infection of over 99% susceptible hosts, even if the number of initially infected hosts is boosted by an incomplete hit-list of size up to 15% of the number of all susceptible hosts.

Chapter 5 The Logic Matrix Approach to Propagation Modelling of Peer-to-Peer Worms

At the beginning of this chapter, I extend definition of a matrix to allow its elements to be variables or constants of logic type; and term such kind of matrices logic matrices. Several operations of logic matrices are defined. Then, topology, state, vulnerability status and quarantine status of a network are represented by its topology logic matrix, state logic matrix, vulnerability logic matrix, and quarantine logic matrix, respectively. Finally, an innovative logic matrix formulation of the propagation process of P2P worms under three different conditions is derived from first principle.

5.1 Logic Matrix and Its Operations

I extend the definition of matrix to allow variables or constants of logic type as its elements and term such kind of matrix logic matrix. The values of variables of logic type can only be one of the two constants of logic type: True (denoted by ‘T’) or False (denoted by ‘F’). If a logic matrix has only one row or one column, we can also term it row logic vector or column logic vector, respectively.

I define absolute value of a variable l of logic type (denoted by $|l|$) as 1 when its value is ‘T’, and 0 when ‘F’; and define absolute value of a logic matrix L (denoted by $|L|$) as the total number of its elements with value ‘T’. According to the above definitions, the absolute value of a logic matrix L can be worked out by summing the absolute value of its each element l, i.e.,

$$|L| = \sum |l| . \tag{5.1}$$

A logic matrix L can be inverted. The resultant $\overline{L}$ is a logic matrix of the same dimension with its element l_{inv} being the result of logic NOT operation of the corresponding element l of the logic matrix to be inverted. It can be defined mathematically as follows:

$$l_{inv} = \bar{l} , \tag{5.2}$$

where the bar over l indicates logic NOT operation.

Two logic matrices A and B can be added together if and only if their dimensions are the same, i.e., they have the same number of rows and the same number of columns. The resultant $S = A + B$ is a logic matrix of the same dimension with its element s_{ij} (in the i-th row and the j-th column) being the result of logic OR operation of the corresponding elements a_{ij} and b_{ij} of the two logic matrices to be added together. It can be defined mathematically as follows:

$$s_{ij} = a_{ij} + b_{ij} , \tag{5.3}$$

where the + sign between a_{ij} and b_{ij} indicates logic OR operation.

Mutation law applies to the logic matrix addition defined above.

Two logic matrices A and B can be multiplied element-by-element if and only if their dimensions are the same, i.e., they have the same number of rows and the same number of columns. The resultant $P = AB$ is a logic matrix of the same dimension with its element p_{ij} (in the i-th row and the j-th column) being the result of logic AND operation of the corresponding elements a_{ij} and b_{ij} of the two logic matrices to be multiplied element-by-element. It can be defined mathematically as follows:

$$p_{ij} = a_{ij}b_{ij}, \qquad (5.4)$$

where $a_{ij}b_{ij}$ indicates logic AND operation of a_{ij} and b_{ij}.

Mutation law applies to the logic matrix element-by-element multiplication defined above.

A logic matrix A can be multiplied by another logic matrix B in the manner of traditional matrix multiplication if and only if their inner dimensions are the same, i.e., number of columns of the multiplicand logic matrix (the left one) is equal to number of rows of the multiplier logic matrix (the right one). The resultant $P = AB$ is a logic matrix with the same number of rows as A and the same number of columns as B. I define value of element p_{ij} (in the i-th row and the j-th column) of the product as determined by the following equation:

$$p_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}, \qquad (5.5)$$

where $a_{ik}b_{kj}$ indicates logic AND operation of a_{ik} and b_{kj}, n denotes inner dimensions of the multiplicand and the multiplier logic matrices, and $\sum$ denotes logic OR operation of all resultants of those logic AND operations.

Contrary to logic matrix addition and logic matrix element-by-element multiplication, mutation law does not apply to the logic matrix multiplication in the manner of traditional matrix multiplication defined above.

Now the stage for later discussion has been set. In the next two sections, I will introduce the concepts of a P2P network's topology logic matrix, state logic matrix, vulnerability logic matrix, and quarantine logic matrix, respectively; and derive our innovative logic matrix formulation of the propagation process of P2P worms under three different conditions from first principle.

5.2 The Logic Matrix Representations

According to the traditional directed graph theory, a P2P overlay network can be represented by a directed graph G, with its set of vertices V representing all peers connected to form the network, and its set of directed edges E representing all directed links among these peers. A directed link from peer i to peer j means peer j is a neighbour of peer i, but peer i is not a neighbour of peer j if there does not exist a directed link from peer j to peer i at the same time. A peer is only able to send messages to its neighbours directly.

Topology of a P2P overlay network consisting of n peers can be represented by an n by n square matrix T with its element t_{ij} (in the i-th row and the j-th column) indicating whether there is a directed link from peer i to peer j.

In this thesis, I propose a different approach from that used under the traditional directed graph theory to indicating the existence or not of a directed link. The logic constant 'T' is used to indicate there is a directed link, and the logic constant 'F' to indicate there is not. Therefore, topology of a P2P overlay network consisting of n peers can be represented by an n by n logic square matrix. I term it topology logic matrix of the P2P overlay network.

Each row of the topology logic matrix of a P2P overlay network forms a row logic vector, which is a logic vector representation of outbound links (neighbours) of a particular peer belonging to the network. I call this row logic vector the peer's topology out-degree logic vector. Each column of the topology logic matrix of a P2P overlay network forms a column logic vector, which is a logic vector representation of inbound links of a particular peer belonging to the network. I call this logic column vector the peer's topology in-degree logic vector. For example, the i-th row of a topology logic matrix represents all outbound links (neighbours) of peer i; and the j-th column of the topology logic matrix represents all inbound links of peer j.

It can be easily derived that values of topology in-degree and topology out-degree of each peer belonging to a P2P overlay network equate to the absolute values of the peer's topology in-degree logic vector and topology out-degree logic vector, respectively, which can be worked out by using equation (5.1).

I represent states of all the n peers belonging to the P2P overlay network by a row logic vector S of length n with its element s_j (the j-th element) indicating whether peer j has been infected by the worm and become infectious. The logic constant 'T' is used to indicate a peer has been infected and become infectious, and the logic constant 'F' to indicate it has not. I term the above row logic vector the P2P overlay network's state logic vector. It can be derived that the total number of infected and infectious peers in a P2P overlay network equates to the absolute value of the network's state logic vector, which can be worked out by using equation (5.1).

A healthy peer vulnerable to the worm can be infected by the worm and become infectious. However, a peer which is not vulnerable to the worm will not be infected by the worm and become infectious. I represent vulnerability status of all the n peers in the P2P overlay network by a row logic vector V of length n with its element v_j (the j-th element) indicating whether peer j is vulnerable to the worm. The logic constant ‘T’ is used to indicate a peer is vulnerable to the worm, and the logic constant ‘F’ to indicate it is not. I term the above logic row vector the P2P overlay network’s vulnerability logic vector. It can be derived that the total number of peers vulnerable to the worm in a P2P overlay network equates to the absolute value of the network’s vulnerability logic vector by using equation (5.1).

I represent quarantine status of all the n peers belonging to the P2P overlay network by a row logic vector Q of length n with its element q_j (the j-th element) indicating whether peer j has been quarantined for the worm. A quarantined healthy peer will not be infected by the worm; and a quarantined infected and infectious peer will be cured and will not be infected again by the worm. The logic constant ‘T’ is used to indicate a peer has been quarantined, and the logic constant ‘F’ to indicate it has not. I term the above row logic vector the P2P overlay network’s quarantine logic vector. It can be derived that the total number of quarantined peers in a P2P overlay network equates to the absolute value of the network’s quarantine logic vector, which can be worked out by using equation (5.1).

5.3 The Logic Matrix Formulation

Based on the above extensions to matrix and its operations and extensions to the matrix representation of a network in the traditional directed graph theory, I am now ready to derive our innovative logic matrix formulation of the propagation process of P2P worms. The derivation is based on the following assumptions.

An infected and infectious peer will send the worm packets to all other peers belonging to the same P2P overlay network to which it has a outbound link, regardless of the state (infected by the worm and infectious or not) and the quarantine status (quarantined for the worm or not) of those peers. A healthy (not infected by the worm and not infectious) peer will be infected by the worm and become infectious once it receives the worm packets from an infectious peer, provided the healthy peer is not quarantined for the worm. An infected and infectious peer will remain in that state once it receives the worm packets from an infectious peer, provided the infected and infectious peer is not quarantined for the worm. A healthy peer quarantined for the worm will not be infected by the worm; and an infected and infectious peer quarantined for the worm will be cured and will not be infected again by the worm.

There are a total of n peers belonging to a logical (not physical) P2P overlay network under consideration. Initially, there are a total of I_0 peers which are infected by the worm and infectious.

According to the above assumptions, the logical P2P overlay network's initial state (State 0) can be represented by its initial state logic vector S_0 of length n; and the absolute value of S_0 equates to the total number of peers which are initially infected by the worm and infectious (I_0), i.e.,

$$|S_0| = I_0 \, . \qquad (5.6)$$

Generally, State g of the logical P2P overlay network can be represented by its state logic vector S_g of length n; and the absolute value of S_g equates to the total number of peers which are infected by the worm and infectious at that state (I_g), i.e.,

$$\left|S_g\right| = I_g \, . \tag{5.7}$$

The next state (State g+1) of the logical P2P overlay network can be represented by its state logic vector S_{g+1} of length n; and the absolute value of S_{g+1} equates to the total number of peers which are infected by the worm and infectious at that state (I_{g+1}), i.e.,

$$\left|S_{g+1}\right| = I_{g+1} \, . \tag{5.8}$$

I notice that the logical P2P overlay network's next state represented by its state logic vector S_{g+1} is fully determined by the network's current state represented by its state logic vector S_g, the network's topology represented by its topology logic matrix T, the network's vulnerability status represented by its vulnerability logic vector V, and the network's quarantine status represented by its quarantine logic vector Q.

If all peers are vulnerable to the P2P worm, I find the relationship among S_{g+1}, S_g, T, V, and Q can be described mathematically as follows:

$$S_{g+1} = S_g + S_g T \overline{Q} \, . \tag{5.9}$$

Let S_g^{new} stand for the second term in the above equation (after the + sign), the above equation can be simplified to

$$S_{g+1} = S_g + S_g^{new} \, . \tag{5.10}$$

The term represented by S_g^{new} actually says if at State *g* at least one peer among those peers from which peer *j* has inbound links is infectious, peer *j* will be infected by the worm and become infectious at State *g*+1 provided peer *j* is not quarantined.

Since both S_g and *Q* are row logic vectors of length *n* and *T* is an *n* by *n* square logic matrix, S_g^{new} will be a row logic vector of length *n*. It can be derived that S_g^{new} is a logic vector representation of all those peers that can be infected by the worm at State *g*+1, given the network's state at State *g* represented by its state logic vector S_g, the network's topology represented by its topology logic matrix *T*, and the network's quarantine status represented by its quarantine logic vector *Q*. S_g^{new} may or may not include peer or peers infected by the worm at states prior to State *g*+1. Then, equations (5.9) and (5.10) can be easily derived.

If quarantined is not enforced at all and not all peers are vulnerable to the P2P worm, equation (5.9) will be changed to

$$S_{g+1} = S_g + S_g TV \quad . \qquad (5.11)$$

The term represented by the second term in the above equation (after the + sign) actually says if at State *g* at least one peer among those peers from which peer *j* has inbound links is infectious, peer *j* will be infected by the worm and become infectious at State *g*+1 provided peer *j* is vulnerable to the worm.

If quarantined is not enforced at all and all peers are vulnerable to the P2P worm, equation (5.11) will be simplified to

$$S_{g+1} = S_g + S_g T \quad . \qquad (5.12)$$

Equation (5.12) is also a special case of equation (5.9) when Q is a row logic vector with all its elements being ‘F’.

Equations (5.9), (5.11), and (5.12) are actually discrete-time deterministic propagation models of P2P worms under three different conditions, respectively, written in the form of difference equations of logic matrix.

Starting from some certain state, there will be no newly infected peer to occur and thus actually, the propagation will stop. The state from which the propagation will cease is the earliest state whose state logic vector S_G satisfies the following equation:

$$|S_{G+1}| = |S_G|, \qquad (5.13)$$

where S_{G+1} stands for the state logic vector of the state immediately after the state with state logic vector S_G.

I call the earliest state whose state logic matrix S_G satisfies (13) the final state of the P2P overlay network.

The proposed logic matrix approach essentially translates the propagation processes of P2P worms into a sequence of logic matrix operations.

Chapter 6 Simulation Experiments: Applications of the Logic Matrix Approach

6.1 Evaluation Metrics

Our evaluation metric for attack performance in this chapter is a P2P worm's coverage rate (denoted by c) in a logical P2P overlay network. It is defined as the ratio of number of peers in the network that can be infected by the worm to number of peers in the network that are vulnerable to the worm. It can be worked out by using the following equation:

$$c = \frac{|S_G|}{|V|}, \tag{6.1}$$

where S_G is the state logic vector of the network when the propagation process has just stopped, and V is the vulnerability logic vector to the worm of the network.

One of our evaluation metrics for network-related characteristics of P2P networks in this chapter is vulnerability rate (denoted by v) to a P2P worm of a logical P2P overlay network, which is defined as the ratio of number of peers in the network that are vulnerable to the worm to total number of peers in the network. It can be worked out by using the following equation:

$$v = \frac{|V|}{n}, \tag{6.2}$$

where V is the vulnerability logic vector to the worm of the network, and n is total number of peers in the network.

Vulnerability rate is set to be from 0% to 100% with 20% interval to fully investigate the impact of vulnerability rate on coverage rate. The basis for parameters variation (the 20% interval) in the simulations of this chapter is small enough to reveal the impact of vulnerability rate on coverage rate and large enough to minimise simulation time.

The rest two of our evaluation metrics for network-related characteristics of P2P networks in this chapter are topology out-degree, which refers to the number of logical neighbours maintained by each peer locally; and network size, which refers to total number of peers in a P2P network.

Our defence-related evaluation metric in this chapter is quarantine rate (denoted by q) for a P2P worm of a logical P2P overlay network. It is defined as the ratio of number of peers belonging to the network that are quarantined for the worm to total number of peers belonging to the network; and can be worked out by using the following equation:

$$q = \frac{|Q|}{n}, \tag{6.3}$$

where Q is the quarantine logic vector for the worm of the network and n is total number of peers belonging to the network.

I apply the proposed logic matrix approach in the simulation experiments under the following three different conditions using MathWorks' MATLAB.

6.2 All Peers Being Vulnerable to the P2P Worm and No Quarantine

In this case, I investigate the impacts of the two different topologies, namely the simple random graph topology and the pseudo power law topology on the coverage rate of P2P worms.

6.2.1 The Simple Random Graph Topology

I investigate the impacts of the two parameters, namely the number of initially infected computers belonging to a P2P network and the mean value of topology out-degree of the network, on the coverage rate of P2P worms in the network.

The implementation in MATLAB assumes there are a total of 10,000 peers (computers) belonging to the logical P2P overlay network under consideration. Therefore, the topology of the overlay network is represented by its topology logic matrix, which is a 10,000 by 10,000 square logic matrix; and its initial state is represented by its initial state logic vector, which is a 1 by 10,000 logic matrix (row logic vector). In the experiments conducted for this sub-section, I assume each peer has the same value of topology out-degree. Peers to which each peer has outbound links are randomly selected from all peers except the peer itself belonging to the overlay network, which means I do not allow loop, that is, no peer has an outbound link to itself. Therefore, I call the topology of the overlay network in the experiments conducted for this sub-section the simple random graph topology.

I conduct the experiments with MATLAB under different combinations of values of the number of initially infected computers and the mean value of topology out-degree.

Firstly, I fix the number of initially infected peers (computers) belonging to the overlay network to be 1, and try to find out the impact of mean value of topology out-degree on the coverage rate of P2P worms in the overlay network. The initially infected peer is randomly select from all peers belonging to the overlay network. A total of 5 scenarios listed in Table 1 are investigated. Experiment for each scenario is repeated 100 times. Then, the mean value of coverage rate and coefficient of variation of coverage rate are worked out. Results from the experiments are listed in Table 6.1.

Table 6.1: A list of the experimental results (only 1 initially infected peer randomly selected from all peers)

Mean Value of Topology Out-Degree	**Mean Value of Coverage Rate (%)**	**Coefficient of Variation of Coverage Rate (%)**
1	1.23	54.81
2	79.64	0.68
3	94.08	0.27
4	98.06	0.16
5	99.31	0.09

As shown by Table 6.1, mean value of topology out-degree has great impact on both mean value and coefficient of variation of coverage rate of P2P worms in the overlay network featuring the simple random graph topology. Increase in mean value of topology out-degree results in increase in mean value of coverage rate but decrease in coefficient of variation of coverage rate. When mean value of topology out-degree is increased to 3, mean value of coverage rate is increased to over 90% and its coefficient of variation becomes very small, which indicates 3 is the minimum mean value of topology out-degree which can make a P2P worm be able to infect most peers with very high certainty.

Then, I fix the number of initially infected peers (computers) belonging to the overlay network to be 10/100, and repeat the above experiments. Results from the experiments are listed in Table 6.2.

Table 6.2: A list of the experimental results (a total of 10 /100 initially infected peers randomly selected from all peers)

Mean Value of Topology Out-Degree	**Mean Value of Coverage Rate (%)**		**Coefficient of Variation of Coverage Rate (%)**	
	Initially infected peers=10	**Initially infected peers=100**	**Initially infected peers=10**	**Initially infected peers=100**
1	4.28	13.53	16.22	5.43
2	79.80	80.06	0.63	0.62
3	94.10	94.16	0.27	0.26
4	98.03	98.06	0.15	0.15
5	99.30	99.31	0.08	0.09

Table 6.2 shows similar trends to those shown by Table 6.1, which indicates the impact of number of initially infected peers on the coverage rate of a P2P worm in the overlay network featuring the simple random graph topology is insignificant.

6.2.2 The Pseudo Power Law Topology

Similar to the previous sub-section, I investigate the impacts of the two parameters, namely the number of initially infected computers belonging to a P2P network and the maximum value of topology out-degree of the network, on the coverage rate of P2P worms in the network.

In the experiments conducted for this sub-section, I assume only a very small number (10 in the experiments) of peers have the maximum value of topology out-degree, and all other peers have the minimum value (1 in the experiments) of topology out-degree. Although the distribution of topology out-degree in the experiments does not strictly follow power law, it does have the most important features of power law distribution, namely peers with maximum value of topology out-degree are rare and most peers have minimum value of topology out-degree. Therefore, I call the topology of the overlay network in the experiments conducted for this sub-section the pseudo power law topology.

I conduct our simulation under different combinations of values of the number of initially infected computers and the maximum value of topology out-degree.

Firstly, I fix the number of initially infected peers (computers) belonging to the overlay network to be 1, and try to find out the impact of maximum value of topology out-degree on the coverage rate in the overlay network. The initially infected peer is randomly select from all peers belonging to the overlay network. A total of 5 scenarios are investigated. In the experiments conducted for this sub-section, I assume each peer has either the maximum value of topology out-degree or the minimum value of topology out-degree. Peers to which each peer has outbound links are randomly selected from all peers except the peer itself belonging to the overlay network. Experiment for each scenario is repeated 100 times. Then, the mean value of coverage rate and coefficient of variation of coverage rate are worked out. Results from the experiments are listed in Table 6.3.

Table 6.3: A list of the experimental results (only 1 initially infected peer randomly selected from all peers)

Maximum Value of Topology Out-Degree	Mean Value of Coverage Rate (%)	Coefficient of Variation of Coverage Rate (%)
100	3.17	200.74
1000	13.83	209.20
2000	14.54	226.10

As shown by Table 6.3, when all initially infected peers are randomly selected from all peers, maximum value of topology out-degree has a little impact on both mean value and coefficient of variation of coverage rate of P2P worms in the overlay network featuring the pseudo power law topology. Increase in maximum value of topology out-degree results in a little increase in mean value of coverage rate and a little increase in coefficient of variation of coverage rate as well, which indicates the small gain in coverage rate could be offset by the small loss in certainty. The worm is not able to infect most peers with high certainty.

After that, I fix the number of initially infected peers (computers) belonging to the overlay network to be 10, and repeat the above experiments. Results from the experiments are listed in Table 6.4.

Table 6.4: A list of the experimental results (a total of 10 initially infected peers randomly selected from all peers)

Maximum Value of Topology Out-Degree	**Mean Value of Coverage Rate (%)**	**Coefficient of Variation of Coverage Rate (%)**
100	11.25	79.51
1000	33.06	111.27
2000	36.23	120.07

Table 6.4 shows similar trends (just an insignificantly higher coverage rate and an insignificantly lower coefficient of variation of coverage rate) to those shown by Table 3, which indicates, when all initially infected peers are randomly selected from all peers, the impact of number of initially infected peers on the coverage rate of a P2P worm in the overlay network featuring the pseudo power law topology is insignificant.

Finally, initially infected peers are randomly select from only those peers with maximum topology out-degree and I repeat all of the above experiments described in this sub-section. Results from the experiments are listed in Table 6.5 and Table 6.6.

Table 6.5: A list of the experimental results (only 1 initially infected peer randomly selected from only those peers with maximum topology out-degree)

Maximum Value of Topology Out-Degree	**Mean Value of Coverage Rate (%)**	**Coefficient of Variation of Coverage Rate (%)**
100	20.74	26.65
1000	78.21	11.17
2000	95.33	0.89

Table 6.6: A list of the experimental results (a total of 10 initially infected peers randomly selected from only those peers with maximum topology out-degree)

Maximum Value of Topology Out-Degree	**Mean Value of Coverage Rate (%)**	**Coefficient of Variation of Coverage Rate (%)**
100	38.50	1.53
1000	85.19	0.41
2000	95.94	0.19

As shown by Table 6.5 and Table 6.6, when all initially infected peers are randomly selected from only those peers with maximum topology out-degree, maximum value of topology out-degree has a great impact on both mean value and coefficient of variation of coverage rate of P2P worms in the overlay network featuring the pseudo power law topology. Increase in maximum value of topology out-degree results in increase in mean value of coverage rate but decrease in coefficient of variation of coverage rate.

However, the impact of number of initially infected peers is insignificant. When maximum value of topology out-degree reaches 2,000, the worm is able to infect most peers with very high certainty, regardless of number of initially infected peers.

6.3 Not All Peers Being Vulnerable to the P2P Worm and No Quarantine

6.3.1 Structured P2P Networks

In a structured P2P network, topology out-degree *d* of each peer is a constant. It is characterized by the following probability distribution:

$$\begin{cases} P(d = k) = 1 \\ P(d \neq k) = 0 \end{cases}, \qquad (6.4)$$

where *k* is a constant.

In this section, I only consider structured P2P networks. Therefore, all peers in the network have the same topology out-degree.

The objective is to investigate the impacts of the network-related characteristics (measured by the evaluation metrics: vulnerability rate *v*, topology out-degree *d*, and network size *n*) on a P2P worm's attack performance in structured P2P networks (measured by the evaluation metric: coverage rate *c*).

6.3.2 Simulation Experiments

Our simulation experiments include scenarios with vulnerability rate of 1.0. Experimental results from them set the benchmark to compare to. When all peers are vulnerable to the worm, equation (5.12) instead of equation (5.11) forms the foundation of our implementation of the proposed logic matrix approach. Otherwise, our implementation is based on equation (5.11).

Our simulation experiments are based on the following assumptions:

- Topology out-degree (*d*) of each peer in the structured P2P network under consideration strictly follows the probability distribution (6.4). Neighbours of a peer are randomly selected from all other peers except the peer itself.
- Peers vulnerable to the worm are selected randomly from all peers in the network.
- There is only 1 initially infected peer, which is selected randomly from all peers in the network that are vulnerable to the worm.

Based on the above assumptions, I first populate the topology logic matrix of the structured P2P network under consideration by letting the probability that a randomly selected peer has *k* neighbours follow (6.4). Then, I populate the vulnerability logic vector of the network, before populating the initial state logic vector of the network.

I conduct the simulation experiments for the three different sets of scenarios. Each of the simulation experiment is repeated 100 times, and then average values of coverage rate are reported as final results.

For the first set of scenarios, I fix topology out-degree at 3. I let vulnerability rate vary from 1.0 to 0.2 with step size – 0.2; and let network size vary from 1,000 to 10,000 with step size 1,000. A vulnerability rate of 1.0 actually means all peers in the network are vulnerable to the worm. The experimental results from the above set of scenarios are illustrated by Figure 6.1.

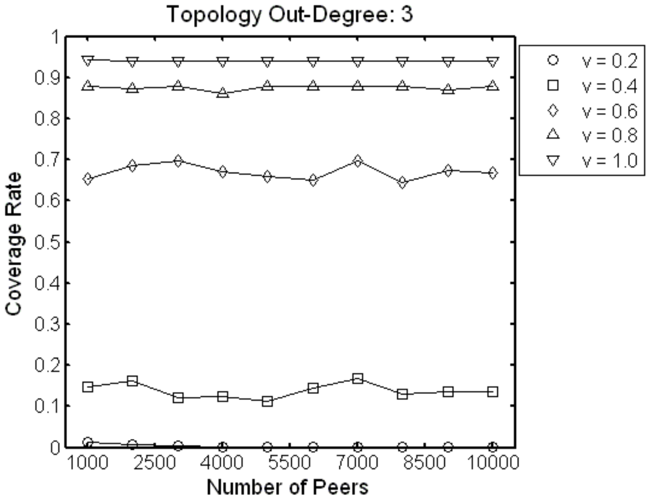


Figure 6.1: Coverage rate as a function of vulnerability rate and network size when topology out-degree is fixed at 3

Figure 6.1 shows that under the set conditions, the coverage rate of a P2P worm in a logical P2P overlay network will decrease if vulnerability rate is decreased. This is sensible since more vulnerable peers in the network naturally lead to higher attack performance measured by the coverage rate. The upper bound of the coverage rate is approximately 0.95. It is achieved when all peers are vulnerable, i.e., v = 1.0 (the top curve in Figure 6.1). The lower bound of the coverage rate is close to 0. It is achieved when 20% peers are vulnerable, i.e., $v = 0.2$ (the bottom curve in Figure 6.1). The coverage rate drops significantly from above 0.6 to below 0.2 when vulnerability rate is decreased from 0.6 to 0.4. The above findings imply that both attackers and defenders can manipulate vulnerability rate v to improve or worsen attack performance, respectively, and more importantly that limiting vulnerability rate to be below 0.4 is critical to defenders.

Besides, Figure 6.1 also shows that, when network size is in the range 1,000-10,000 inclusive, it has no significant impact on attack performance measured by the coverage rate if both topology out-degree and vulnerability rate are fixed. This finding reveals that neither attackers nor defenders can manipulate network size n to improve or worsen attack performance, respectively. It also implies that I can choose a smaller value in the range 1,000-10,000 for network size n in our later experiments to shorten simulation time.

For the second set of scenarios, I fix vulnerability rate at 0.5. I let topology out-degree vary from 1 to 5 with step size 1; and let network size vary from 1,000 to 10,000 with step size 1,000. The experimental results from the above set of scenarios are illustrated by Figure 6.2.

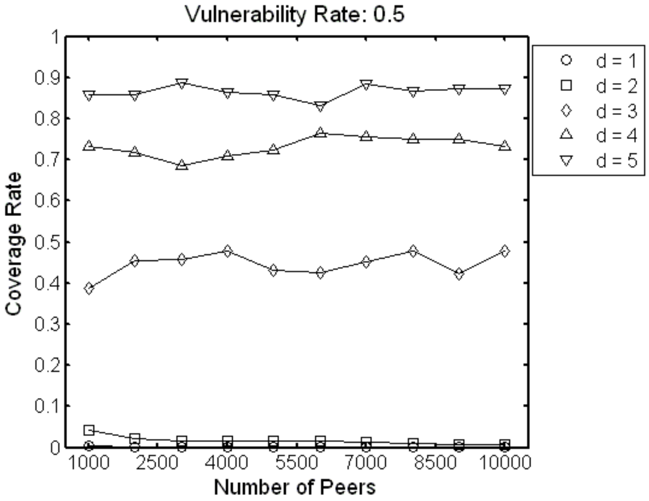


Figure 6.2: Coverage rate as a function of topology out-degree and network size when vulnerability is fixed at 0.5

Figure 6.2 shows that under the set conditions, the coverage rate of a P2P worm in a logical P2P overlay network will increase if topology out-degree is increased. This is sensible since more neighbours a peer in the network has naturally lead to higher attack performance measured by the coverage rate. The upper bound of the coverage rate is approximately 0.85. It is achieved when all peers have 5 neighbours, i.e., *d* = 5 (the top curve in Figure 6.2). The lower bound of the coverage rate is 0. It is achieved when all peers have only 1 neighbour, i.e., *d* = 1 (the bottom curve in Figure 6.1). The coverage rate drops significantly from above approximately 0.4 to below 0.05 when topology out-degree decreased from 3 to 2. The above findings imply that both attackers and defenders can manipulate topology out-degree *d* to improve or worsen attack performance, respectively, and more importantly that limiting topology out-degree to be below or equal to 2 is critical to defenders.

Based on the common finding from our first 2 sets of simulation experiments that when network size is in the range 1,000-10,000 inclusive, it has no significant impact on attack performance, for the third set of scenarios, I fix network size at 5,000 to shorten simulation time. I investigate the two cases given below:

Case 1 - In this case, I let vulnerability rate vary from 0.1 to 1.0 with step size 0.1; and let topology out-degree vary from 1 to 5 with step size 1. Here, our focus is on the impact of vulnerability rate rather than topology out-degree on attack performance measured by the coverage rate. Therefore, I choose a smaller step size for vulnerability rate, but only a few topology out-degree values are investigated.

The experimental results from the above case are illustrated by Figure 6.3. Figure 6.3 shows that generally the coverage rate of a P2P worm in a logical P2P overlay network will increase if vulnerability rate is increased. This is sensible since more vulnerable peers in the network naturally lead to higher attack performance measured by the coverage rate.

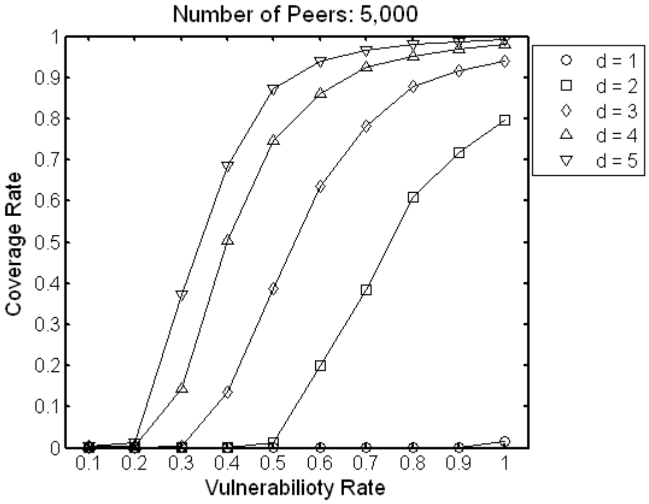


Figure 6.3: Coverage rate as a function of topology out-degree and vulnerability rate when network size is fixed at 5,000 (Case 1)

More importantly, Figure 6.3 also shows that the takeoff points on the curves do not correspond to the same value of vulnerability rate. Here, takeoff point refers to the point on a curve in Figure 6.3 immediately to the right of which the slope of the curve increases dramatically. For instance, when topology out-degree is fixed at 5, the takeoff point corresponds to vulnerability rate 0.2; and when topology out-degree is reduced to 3, the takeoff point corresponds to vulnerability rate 0.3. Generally, the corresponding vulnerability rate will increase if topology out-degree is reduced. This is understandable since fewer neighbours demand more vulnerable peers to achieve the same attack performance. It can be found from Figure 6.3 that 0.2 is a critical value of vulnerability rate since if vulnerability rate is below that value, the worm cannot propagate successfully in the network.

Case 2 - In this case, I let topology out-degree vary from 1 to 10 with step size 1; and let vulnerability rate vary from 0.2 to 1.0 with step size 0.2. Here, our focus is on the impact of topology out-degree rather than vulnerability rate on attack performance measured by the coverage rate. Therefore, a large range of topology out-degree values are investigated, but I choose a larger step size for vulnerability rate.

The experimental results from the above case are illustrated by Figure 6.4. Figure 6.4 shows that generally the coverage rate of a P2P worm in a logical P2P overlay network will increase if topology out-degree is increased. This is sensible since more neighbours a peer in the network has naturally lead to higher attack performance measured by the coverage rate.

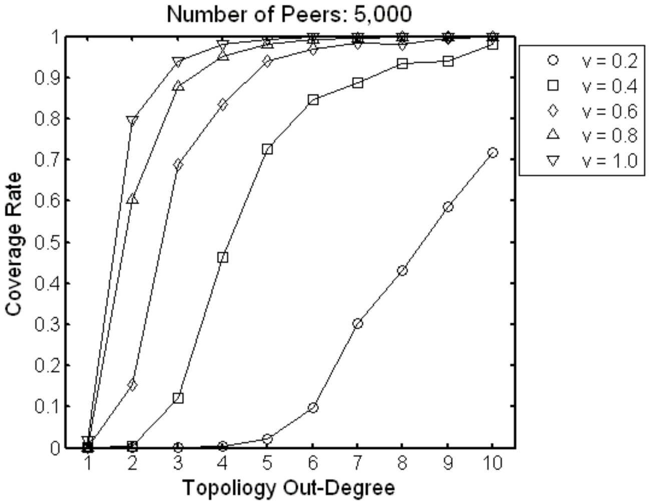


Figure 6.4: Coverage rate as a function of vulnerability rate and topology out-degree when network size is fixed at 5,000 (Case 2)

More importantly, Figure 6.4 also shows that the takeoff points on the curves do not correspond to the same value of topology out-degree. When vulnerability rate is fixed at 1.0, the takeoff point corresponds to topology out-degree 1; and when vulnerability rate is reduced to 0.2, the takeoff point corresponds to topology out-degree 5. Generally, the corresponding topology out-degree will increase if vulnerability rate is reduced. This is understandable since fewer vulnerable peers demand more neighbours a peer in the network has to achieve the same attack performance.

6.4 All Peers Being Vulnerable to the P2P Worm and Quarantine Being Existent

6.4.1 Unstructured P2P Networks

In an unstructured P2P network, topology out-degree (*d*) of each peer is a variable. It is characterized by the following power law distribution:

$$\begin{cases} D_{\min} \le k \le D_{\max} \\ P(d = k) = \dfrac{C}{k^A} \\ P(d \ne k) = 0 \end{cases}, \qquad (6.5)$$

where D_{min} and D_{max} stands for minimum topology out-degree and maximum topology out-degree, respectively, A represents power law degree, and C is a constant. Set of equations (6.5) gives the probability that a randomly selected peer has k neighbours.

In this section, I only consider unstructured P2P networks. Therefore, not all peers in the network have the same topology out-degree.

Our paramount objective is to find a quarantine tactic whose enforcement will lead to a lower attack performance (measured by the attach-related evaluation metric: coverage rate *c*) at a lower cost of defence effort (measured by the defence-related evaluation metric: quarantine rate *q*).

According to probability theory, the following equations must hold:

$$1 = \sum_{k=D_{\min}}^{D_{\max}} P(d = k) = C \sum_{k=D_{\min}}^{D_{\max}} \frac{1}{k^{A}} , \tag{6.6}$$

$$E(d) = \sum_{k=D_{\min}}^{D_{\max}} kP(d = k) = C \sum_{k=D_{\min}}^{D_{\max}} \frac{1}{k^{A-1}} , \tag{6.7}$$

where *E*(*d*) stands for expected value of topology out-degree.

Then, it can be easily derived from equations (6.6) and (6.7) that power law degree *A* is a function of D_{min}, D_{max}, and *E*(*d*) described implicitly by the following equation:

$$E(d) = \frac{\sum_{k=D_{\min}}^{D_{\max}} \frac{1}{k^{A-1}}}{\sum_{k=D_{\min}}^{D_{\max}} \frac{1}{k^{A}}} . \tag{6.8}$$

Finally, once power law degree *A* is determined according to equation (6.8) given D_{min}, D_{max}, and *E*(*d*), the constant *C* can be worked out according to equation (6.6) or equation (6.7).

The most important feature of the above power law distribution of topology out-degree in the unstructured P2P system is that there are fewer peers with larger topology out-degree than those with smaller topology out-degree.

Let D_{min} = 1, and D_{max} vary from 100 to 1,000 and expected value of topology out-degree $E(d)$ vary from 2 to 32, I numerically determine power law degree A. The results are shown in Figure 6.5.

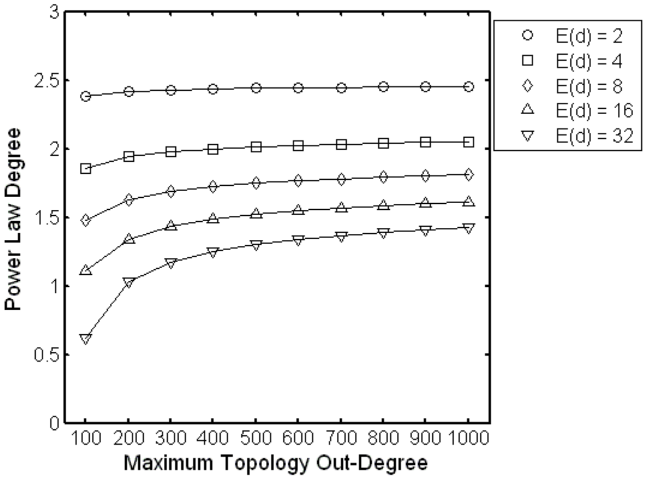


Figure 6.5: Power law degree as a function of maximum topology out-degree and expected value of topology out-degree given minimum topology out-degree being 1

Figure 6.5 shows that a larger maximum topology out-degree requires a larger power law degree, and that a larger expected value of topology out-degree demands a smaller power law degree.

6.4.2 Simulation Experiments

Our simulation experiments are based on the following assumptions:

- Topology out-degree (d) of each peer belonging to the unstructured P2P network under consideration strictly follows the power law distribution (6.5). $E(d)$ = 3,

D_{min} = 1, and D_{max} varies from 100 to 1,000 with step size 100. Neighbours of a peer are randomly selected from all other peers except the peer itself.

- Peers quarantined are selected accordingly based on the quarantine tactics enforced, which are detailed in the next two subsections.
- There is only 1 initially infected peer, which is selected randomly from all peers not quarantined.
- I conduct the simulation experiments for the two different values of *n* (total number of peers belonging to the system). I first assume *n* to be 5,000 and then double it, i.e., assume *n* to be 10,000. I believe 10,000 peers are sufficient for our simulation experiments, and intend to investigate whether 5,000 peers will generate significantly different results.

Based on the above assumptions, I populate the topology logic matrix of the unstructured P2P network under consideration by letting the probability that a randomly selected peer has *k* neighbours follow set of equations (6.5). How to populate the quarantine logic vector of the network is detailed later. Once it is populated, I can populate the initial state logic vector of the network.

Our simulation experiments include scenarios with no quarantine at all. Experimental results from them set the benchmark to compare to. When there is no quarantine, equation (5.12) instead of equation (5.9) forms the foundation of our implementation of the proposed logic matrix approach. When quarantine is enforced, our implementation is based on equation (5.9).

I conduct the simulation experiments for the two different quarantine tactics, namely random quarantine and larger topology out-degree priority quarantine. Each of our simulation experiment is repeated 100 times, and then average values of coverage rate are reported as final results.

I. Random Quarantine

Random quarantine means peers quarantined are randomly selected from all peers. I populate the quarantine logic vector of the unstructured P2P network under consideration by letting each peer have the same probability of being quarantined when this quarantine tactic is enforced. Then, I populate the initial state logic vector of the network.

I conduct the experiments for the 5 sets of scenarios with quarantine rate q varying from 0 to 0.4 with step size 0.1. A quarantine rate of 0 actually means no quarantine at all. I include no quarantine as a special case of random quarantine, which facilitates comparison of experimental results.

The experimental results from random quarantine are illustrated by Figure 6.6 and Figure 6.7 for the two cases: n (total number of peers belonging to the P2P network) = 5,000 and n = 10,000, respectively.

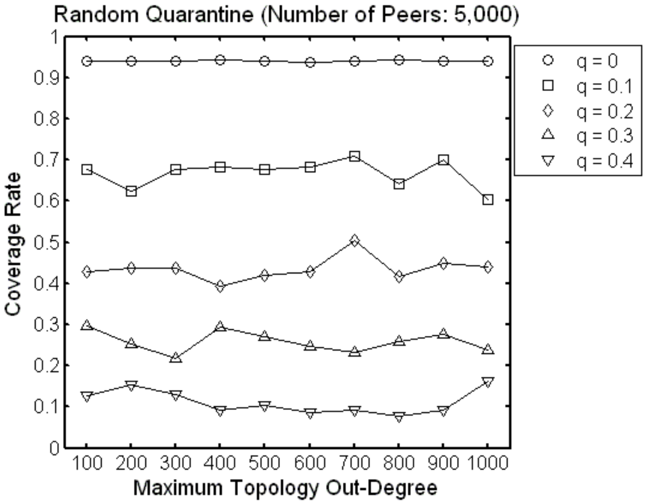


Figure 6.6: Coverage rate under random quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 5,000 peers in the P2P system

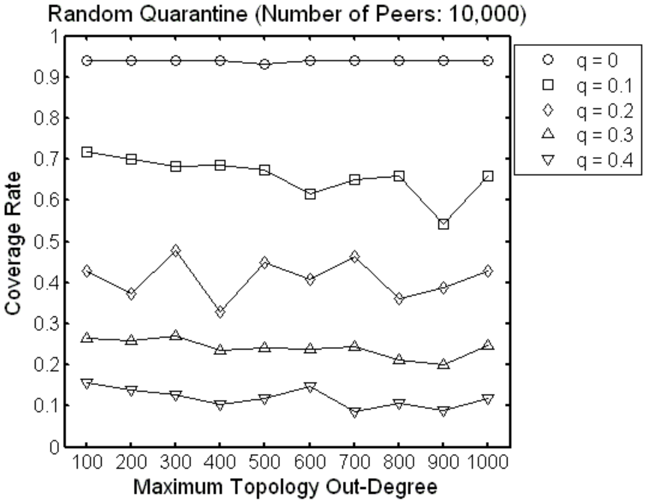


Figure 6.7: Coverage rate under random quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 10,000 peers in the P2P system

Figure 6.6 and Figure 6.7 show that generally, coverage rate of a P2P worm in a logical P2P overlay network will decrease if quarantine rate is increased. This is sensible because a higher defence effort will naturally lead to a lower attack performance. However, as mentioned previously, our paramount objective is to find a quarantine tactic whose enforcement will lead to a lower attack performance at a lower cost of defence effort. Therefore, the above finding cannot serve our paramount objective. Besides, Figure 6.6 and Figure 6.7 also show that maximum topology out-degree has no significant impact on attack performance and defence effort when it is in the range 100-1,000 inclusive, and that 5,000 peers will not generate significantly different results.

The above findings reveal that neither attackers nor defenders can manipulate n or D_{max} to improve attack performance or reduce defence effort, respectively. They also imply that I can choose the smallest value of D_{max} (100) and the smaller value of n (5,000) in our future experiments to shorten simulation time.

II. Larger Topology Out-Degree Priority Quarantine

Larger topology out-degree priority quarantine means peers with larger topology out-degree are quarantined prior to peers with smaller topology out-degree.

When this quarantine tactic is enforced, I populate the quarantine logic vector of the unstructured P2P network under consideration by following the procedure given below.

Firstly, I work out absolute value of each peer's topology out-degree logic vector. Secondly, all peers are sorted in descending order of the absolute value calculated above. By doing this, I actually sort all peers into a list in descending order of number of neighbours since, as mentioned previously, each peer's topology out-degree logic vector is a logic vector representation of its outbound links (neighbours). Thirdly, I quarantine peers in the same order as their order in the sorted list of peers. Then, I populate the initial state logic vector of the network.

I conduct the experiments for the 5 sets of scenarios with quarantine rate *q* varying from 0 to 0.16 (40% of 0.4, which is the maximum quarantine rate investigated under random quarantine) with step size 0.04.

The experimental results from larger topology out-degree priority quarantine are illustrated by Figure 6.8 and Figure 6.9 for the two cases: *n* (total number of peers belonging to the P2P network) = 5,000 and *n* = 10,000, respectively.

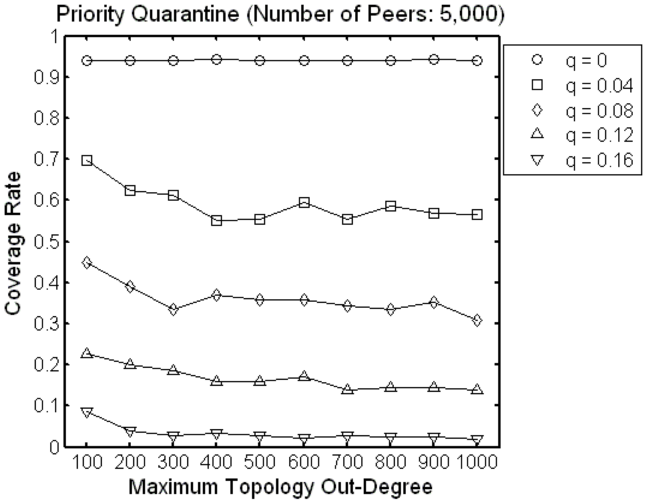


Figure 6.8: Coverage rate under priority quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 5,000 peers in the P2P system

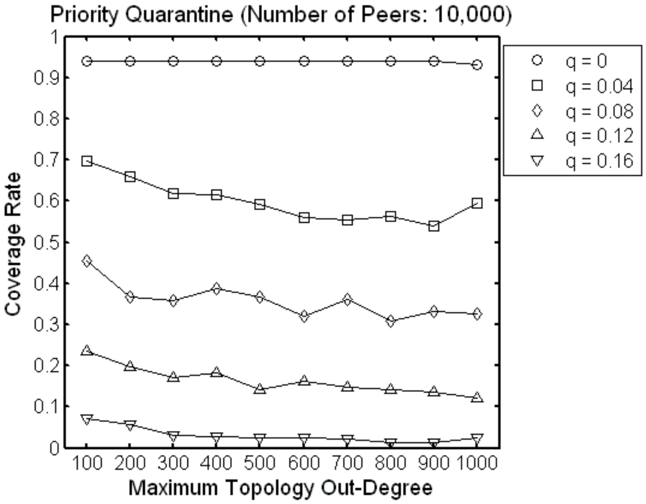


Figure 6.9: Coverage rate under priority quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 10,000 peers in the P2P system

Figure 6.8 and Figure 6.9 show that generally, coverage rate of a P2P worm in a logical P2P overlay network will decrease if quarantine rate is increased. Besides, Figure 6.8 and Figure 6.9 also show that maximum topology out-degree has no significant impact on attack performance and defence effort when it is in the range 100-1,000 inclusive, and that 5,000 peers will not generate significantly different results. The above findings are the same as those from random quarantine.

If I compare the bottom curve in Figure 6.6 to the bottom curve in Figure 6.8, it can be seen that larger topology out-degree priority quarantine demands a lower defence effort (quarantine rate q = 0.16) to achieve a lower attack performance (coverage rate c < 0.1), and that random quarantine demands a higher defence effort (quarantine rate q = 0.4) to achieve a higher attack performance (coverage rate c < 0.2). The same result as above can be found if I compare the bottom curve in Figure 6.7 to the bottom curve in Figure

6.9. The above finding exactly serves our paramount objective, which is to find a quarantine tactic whose enforcement will lead to a lower attack performance at a lower cost of defence effort.

Therefore, according to our experimental results, larger topology out-degree priority quarantine outperforms random quarantine. Larger topology out-degree priority quarantine is exactly the quarantine tactic I am looking for since it demands only 40% (0.16/0.4) defence effort to achieve 50% (0.1/0.2) attack performance, compared to random quarantine. In other words, larger topology out-degree priority quarantine is much more efficient than random quarantine.

Chapter 7 Conclusions

The aim of this research is to establish mathematical models of computer worms, especially P2P worms. This thesis presents a study on modeling and simulating the propagation of computer worms. I present the proposed discrete-time deterministic CFAP model of active worms. I also propose a novel logic matrix approach to modelling the propagation of P2P worms, which are essentially discrete-time deterministic propagation models of P2P worms. The approach's ease of employment is demonstrated by its applications in the simulation experiments.

Implications of this research for practitioners as well as implications to the broader security literature include that P2P worms are not only much more difficult to model but also much more difficult to defend than non-P2P worms, both due to their different propagation mechanism from non-P2P worms. Therefore, future offenders are more likely to employ P2P worms rather than non-P2P worms as a tool to launch their attacks.

7.1 Major Contributions

The major contributions of this thesis are as follows.

- It was found that uniform scanning is an indispensable elementary target discovery technique of active worms. This point is of extreme importance when multiple target discovery techniques are to be employed, which means uniform scanning must be included as one of those target discovery techniques to be employed.

- I found the combination of target discovery techniques that can best accelerate the propagation of active worms.
- I proposed a discrete-time deterministic CFAP model of active worms.
- It was derived from mathematical analysis that in order to accelerate an active worm’s propagation, we must try to let the active worm infect the first susceptible hosts and enter its fast spread phase as soon as possible. This point gives guidance to how to best accelerate an active worm’s propagation.
- I proposed several strategies to shorten an active worm’s slow start phase in its propagation, and found the cost-effective hit-list size and average size of internally generated target lists based on the cost and benefit analysis.
- I proposed a novel logic matrix approach to modelling the propagation of P2P worms by modelling the propagation processes of P2P worms by difference equations of logic matrix.
- I found the impacts of the two different topologies, namely structured and unstructured P2P networks, on a P2P worm’s attack performance; and compared the effects of two different quarantine tactics, namely random quarantine and priority quarantine.

7.2 Limitations

In our models, temporal issues, such as the time lag for P2P worms to infect peers and the time spent in quarantining peers, have not been considered since the paramount objective of these models is to facilitate determining the maximum number of peers in a P2P system that can be infected.

7.3 Future Work

To make it more practical to accommodate the dynamic P2P network where peers can join and leave a network, a P2P network's topology logic matrix needs to be updated once a peer joins or leaves the network, which means the topology logic matrix of the P2P network is constantly changing. In the future, I am going to incorporate the above idea in the simulation experiments.

Besides, how to employ the combination of target discovery techniques to accelerate P2P worm propagation using the logic matrix representation could be a potential topic of future research.

References

[1] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A Taxonomy of Computer Worms," in *WORM '03,* Washington D.C., USA, 2003, pp. 11-18.

[2] X. Fan and Y. Xiang, "Defending against the propagation of active worms," in *Proceedings of the 5th International Conference on Embedded and Ubiquitous Computing,* Shanghai, China: Los Alamitos, California.: IEEE Computer Society, 2008, pp. 350-355.

[3] X. Fan and Y. Xiang, "Defending against the propagation of active worms," *Journal of supercomputing,* vol. 51, pp. 167-200, 2010.

[4] X. Fan, W. W. Guo, and M. Looi, "Modeling and simulating the propagation of unstructured peer-to-peer worms," in *2011 Seventh International Conference on Computational Intelligence and Security,* Sanya, China: Los Alamitos, California.: IEEE Computer Society, 2011, pp. 573-577.

[5] X. Fan and Y. Xiang, "Modeling the propagation of peer-to-peer worms," *Future generation computer systems,* vol. 26, (2010), pp. 1433-1443, 2010.

[6] X. Fan and Y. Xiang, "Modeling the propagation process of topology-aware worms," in *Network and parallel computing: 2009 Sixth IFIP International Conference on Network and Parallel Computing (NPC 2009),* Gold Coast, Australia: Los Alamitos, California.: IEEE Computer Society, 2009, pp. 182-189.

[7] X. Fan and Y. Xiang, "Propagation modeling of peer-to-peer worms," in *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA),* Perth, Australia: Los Alamitos, California: IEEE Computer Society, 2010, pp. 1128-1135.

[8] X. Fan and Y. Xiang, "Modeling the propagation of peer-to-peer worms under quarantine," in *2010 IEEE/IFIP Network Operations and Management Symposium 2010,* Osaka, Japan: Piscataway, NJ: IEEE, 2010, pp. 942-945.

[9] D. Moore, C. Shannon, and J. Brown, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," in *IMW '02*, Marseille, France, 2002, pp. 273-284.

[10] C. C. Zou, D. Towsley, and W. Gong, "On the Performance of Internet Worm Scanning Strategies," University of Massachusetts Technical Report: TR-03-CSE-07, 2003.

[11] C. C. Zou, D. Towsley, W. Gong, and S. Cai, "Routing Worm: A Fast, Selective Attack Worm Based on IP Address Information," in *PADS '05*, 2005, pp. 199-206.

[12] Z. Chen and C. Ji, "Importance-Scanning Worm Using Vulnerable-Host Distribution," in *IEEE GLOBECOM*, 2005, pp. 1779-1784.

[13] Z. Chen and C. Ji, "A Self-Learning Worm Using Importance Scanning," in *WORM '05*, Fairfax, VA, USA, 2005, pp. 22-29.

[14] S. Staniford, V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time," in *Security '02*, San Francisco, CA, USA, 2002, pp. 149-167.

[15] S. Staniford, D. Moore, V. Paxson, and N. Weaver, "The Top Speed of Flash Worms," in *WORM '04*, Washington D.C., USA, 2004, pp. 33-42.

[16] C. C. Zou, W. Gong, and D. Towsley, "Code Red Worm Propagation Modeling and Analysis," in *CCS '02*, Washington D.C., USA, 2002, pp. 138-147.

[17] R. M. Anderson and R. M. May, *Infectious Diseases of Humans: Dynamics and Control.* Oxford: Oxford University Press, 1991.

[18] H. Andersson and T. Britton, *Stochastic Epidemic Models and Their Statistical Analysis.* New York: Springer-Verlag, 2000.

[19] N. T. Bailey, *The Mathematical Theory of Infectious Diseases and Its Applications*. New York: Hafner Press, 1975.

[20] J. C. Frauenthal, *Mathematical Modeling in Epidemiology*. New York: Springer-Verlag, 1980.

[21] D. J. Daley and J. Gani, *Epidemic Modelling: An Introduction*. Cambridge: Cambridge University Press, 1999.

[22] Z. Chen, L. Gao, and K. Kwiat, "Modeling the Spread of Active Worms," in *IEEE INFOCOM*, 2003, pp. 1890-1900.

[23] K. Rohloff and T. Basar, "Stochastic Behavior of Random Constant Scanning Worms," in *14th ICCCN*, San Diego, CA, USA, 2005, pp. 339-344.

[24] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Probability Theory*. Boston: Houghton Mifflin, 1971.

[25] S. Sellke, N. B. Shroff, and S. Bagchi, "Modeling and Automated Containment of Worms," in *DSN '05*, 2005, pp. 528-537.

[26] S. Karlin and H. M. Taylor, *A First Course in Stochastic Processes*, 2nd ed.: Academic Press, 1975.

[27] S. Ross, *Stochastic Processes*, 2nd ed.: John Wiley & Sons, 1996.

[28] Y. Wang and C. Wang, "Modeling the Effects of Timing Parameters on Virus Propagation," in *WORM '03*, Washington D.C., USA, 2003, pp. 61-66.

[29] Rüdiger Schollmeier, A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications, Proceedings of the First International Conference on Peer-to-Peer Computing, IEEE (2002).

[30] Ahson, Syed A. & Ilyas, Mohammad, ed. (2008). SIP Handbook: Services, Technologies, and Security of Session Initiation Protocol. Taylor & Francis. p. 204. ISBN 9781420066043.

[31] Zhu, Ce et al., ed. (2010). Streaming Media Architectures: Techniques and Applications: Recent Advances. IGI Global. p. 265. ISBN 9781616928339.

[32] Kamel, Mina et al. (2007). "Optimal Topology Design for Overlay Networks". In Akyildiz, Ian F. Networking 2007: Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet: 6th International IFIP-TC6 Networking Conference, Atlanta, GA, USA, May 14-18, 2007 Proceedings. Springer. p. 714. ISBN 9783540726050.

[33] Filali, Imen et al. (2011). "A Survey of Structured P2P Systems for RDF Data Storage and Retrieval". In Hameurlain, Abdelkader et al. Transactions on Large-Scale Data- and Knowledge-Centered Systems III: Special Issue on Data and Knowledge Management in Grid and PSP Systems. Springer. p. 21. ISBN 9783642230738.

[34] Zulhasnine, Mohammed et al. (2013). "P2P Streaming Over Cellular Networks: Issues, Challenges, and Opportunities". In Pathan et al. Building Next-Generation Converged Networks: Theory and Practice. CRC Press. p. 99. ISBN 9781466507616.

[35] Shen, Xuemin; Yu, Heather; Buford, John; Akon, Mursalin (2009). Handbook of Peer-to-Peer Networking (1st ed.). New York: Springer. p. 118. ISBN 0-387-09750-3.

[36] Chervenak, Ann & Bharathi, Shishir (2008). "Peer-to-peer Approaches to Grid Resource Discovery". In Danelutto, Marco et al. Making Grids Work: Proceedings of the CoreGRID Workshop on Programming Models Grid and P2P System Architecture Grid Systems, Tools and Environments 12-13 June 2007, Heraklion, Crete, Greece. Springer. p. 67. ISBN 9780387784489.

[37] Jin, Xing & Chan, S.-H. Gary (2010). "Unstructured Peer-to-Peer Network Architectures". In Shen et al. Handbook of Peer-to-Peer Networking. Springer. p. 119. ISBN 978-0-387-09750-3.

[38] Lv, Qin et al. (2002). "Can Heterogenity Make Gnutella Stable?". In Druschel, Peter et al. Peer-to-Peer Systems: First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers. Springer. p. 94. ISBN 9783540441793.

[39] Typically approximating O(log N), where N is the number of nodes in the P2P system[citation needed]

[40] Other design choices include overlay rings and d-Torus. See for example Bandara, H. M. N. D; A. P. Jayasumana (2012). "Collaborative Applications over Peer-to-Peer Systems – Challenges and Solutions". Peer-to-Peer Networking and Applications. doi:10.1007/s12083-012-0157-3.

[41] R. Ranjan, A. Harwood, and R. Buyya, "Peer-to-peer based resource discovery in global grids: a tutorial," IEEE Commun. Surv., vol. 10, no. 2. and P. Trunfio, "Peer-to-Peer resource discovery in Grids: Models and systems," Future Generation Computer Systems archive, vol. 23, no. 7, Aug. 2007.

[42] Kelaskar, M.; Matossian, V.; Mehra, P.; Paul, D.; Parashar, M. (2002), A Study of Discovery Mechanisms for Peer-to-Peer Application

[43] Dabek, Frank; Ben Zhao, Peter Druschel, John Kubiatowicz and Ion Stoica (2003). "Towards a Common API for Structured Peer-to-Peer Overlays". Peer-to-Peer Systems II. Lecture Notes in Computer Science 2735: 33–44. doi:10.1007/978-3-540-45172-3_3.

[44] Moni Naor and Udi Wieder. Novel Architectures for P2P Applications: the Continuous-Discrete Approach. Proc. SPAA, 2003.

[45] Gurmeet Singh Manku. Dipsea: A Modular Distributed Hash Table. Ph. D. Thesis (Stanford University), August 2004.

[46] Li, Deng et al. (2009). Vasilakos, A.V. et al., ed. Autonomic Communication. Springer. p. 329. ISBN 978-0-387-09752-7.

[47] Bandara, H. M. N. Dilum; Anura P. Jayasumana (January 2012). "Evaluation of P2P Resource Discovery Architectures Using Real-Life Multi-Attribute Resource and Query Characteristics". IEEE Consumer Communications and Networking Conf. (CCNC '12).

[48] Ranjan, Rajiv; Harwood, Aaron; Buyya, Rajkumar (1 December 2006), A Study on Peer-to-Peer Based Discovery of Grid Resource Information

[49] Ranjan, Rajiv; Chan, Lipo; Harwood, Aaron; Karunasekera, Shanika; Buyya, Rajkumar. "Decentralised Resource Discovery Service for Large Scale Federated Grids" (PDF).

[50] Sorkin, Andrew Ross (4 May 2003). "Software Bullet Is Sought to Kill Musical Piracy". New York Times. Retrieved 5 November 2011.

[51] P. Antoniadis and B. Le Grand, "Incentives for resource sharing in self-organized communities: From economics to social psychology," Digital Information Management (ICDIM '07), 2007

[52] Y. Xiang, X. Fan, and W. T. Zhu, "Propagation of active worms: a survey," *International journal of computer systems science & engineering*, vol. 24, pp. 157-172, 2009.

[53] X. Fan and Y. Xiang, "Accelerating the propagation of active worms by employing multiple target discovery techniques," in *Lecture notes in computer science*. vol. 5245/2008 J. C. e. al, Ed. Berlin: Springer Verlag, 2008, pp. 150 -161.

[54] D. Ellis, "Worm Anatomy and Model," in *WORM '03*, Washington D.C., USA, 2003, pp. 42-50.

[55] C. C. Zou, L. Gao, W. Gong, and D. Towsley, "Monitoring and Early Warning for Internet Worms," in *CCS '03*, Washington D.C., USA, 2003, pp. 190-199.

[56] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The Monitoring and Early Detection of Internet Worms," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 961-974, 2005.

[57] A. Wagner and T. Dubendorfer, "Experiences with Worm Propagation Simulations," in *WORM '03*, Washington D.C., USA, 2003, pp. 34-41.

[58] N. Weaver, I. Hamadeh, G. Kesidis, and V. Paxson, "Preliminary Results Using Scale-Down to Explore Worm Dynamics," in *WORM '04*, Washington D.C., USA, 2004, pp. 65-72.

[59] X. Fan and Y. Xiang, "Shortening the slow start phase in the propagation of active worms," in *CSA 2008: Proceedings of International Symposium on Computer Science and Its Applications*, Hobart, Australia: Los Alamitos, California.: IEEE Computer Society, 2008, pp. 90-95.

Appendices

Refereed Journal Articles by the Candidate

X. Fan and Y. Xiang, "Modeling the propagation of peer-to-peer worms," *Future generation computer systems,* vol. 26, (2010), pp. 1433-1443, 2010. **(ERA Tier A Journal)**

X. Fan, W. W. Guo, and M. Looi, "Modeling and simulating the propagation of structured peer-to-peer worms," *Journal of Networks*. **(Accepted) (ERA Tier A Journal)**

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

Volume 26, issue 8, October 2010
ISSN 0167-739X
ELSEVIER
THE INTERNATIONAL JOURNAL OF
F FUTURE G GENERATION C COMPUTER S SYSTEMS
GRID COMPUTING AND eSCIENCE
Editor-in-Chief:
Peter Sloot
Available online at www.sciencedirect.com
ScienceDirect

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Future Generation Computer Systems 26 (2010) 1433–1443

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Modeling the propagation of Peer-to-Peer worms

Xiang Fan [a], Yang Xiang [b,*]

[a] School of Management and Information Systems, Central Queensland University, Rockhampton, Queensland 4702, Australia
[b] School of Information Technology, Deakin University, Burwood, Victoria 3125, Australia

ARTICLE INFO

Article history:
Received 30 November 2009
Received in revised form
30 March 2010
Accepted 16 April 2010
Available online 24 April 2010

Keywords:
Network security
Invasive software
Worms
Peer-to-Peer
Propagation
Modeling
Defense

ABSTRACT

Propagation of Peer-to-Peer (P2P) worms in the Internet is posing a serious challenge to network security research because of P2P worms' increasing complexity and sophistication. Due to the complexity of the problem, no existing work has solved the problem of modeling the propagation of P2P worms, especially when quarantine of peers is enforced. This paper presents a study on modeling the propagation of P2P worms. It also presents our applications of the proposed approach in worm propagation research.

Motivated by our aspiration to invent an easy-to-employ instrument for worm propagation research, the proposed approach models the propagation processes of P2P worms by difference equations of a logic matrix, which are essentially discrete-time deterministic propagation models of P2P worms. To the best of our knowledge, we are the first using a logic matrix in network security research in general and worm propagation modeling in particular.

Our major contributions in this paper are firstly, we propose a novel logic matrix approach to modeling the propagation of P2P worms under three different conditions; secondly, we find the impacts of two different topologies on a P2P worm's attack performance; thirdly, we find the impacts of the network-related characteristics on a P2P worm's attack performance in structured P2P networks; and fourthly, we find the impacts of the two different quarantine tactics on the propagation characteristics of P2P worms in unstructured P2P networks. The approach's ease of employment, which is demonstrated by its applications in our simulation experiments, makes it an attractive instrument to conduct worm propagation research.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Worms and their variants have been a serious challenge to network security research for many years. Worms can be classified according to the techniques by which they discover new targets to infect. Scanning, which entails probing a set of addresses to identify vulnerable hosts, is the technique most widely employed by worms [1]. Scanning can be implemented differently, which leads to several different types of scanning such as random scanning, localized scanning [2], sequential scanning [3], routable scanning [4], selective scanning [4], importance scanning [5,6], and topological scanning. Topological scanning was employed by the Morris Internet Worm of 1988 as its target discovery technique [7].

Worms employing all other types of scanning, except topological scanning, among the above types do not need to have any knowledge on the topology of the network they intend to propagate across. On the contrary, worms employing topological scanning must have more information on the network they intend to propagate over, or have the capability to discover that information if they do not have it in advance. Therefore, worms employing topological scanning are also called topology-aware worms.

Typical examples of topology-aware worms are worms attacking a flaw in a Peer-to-Peer (P2P) application and propagating across the P2P network by getting lists of peers from their victims and directing their subsequent attacks to those peers. This sort of topology-aware worms are called P2P worms. The Slapper worm of 2003 was a typical example of P2P worms [8]. The subsequent appearance of variations of the Slapper worm (the Slapper.B worm a.k.a. Cinik and the Slapper.C worm a.k.a. Unlock) indicates that P2P worms are becoming increasingly complex and sophisticated [8].

Due to the recent popularity of P2P systems with their increasing number of users, they have become the most effective vehicles for topology-aware worms to achieve the fastest propagation across the Internet. Propagation of P2P worms on top of P2P systems can result in devastating damage, as illustrated by [9]. P2P worms are posing a serious challenge to the Internet.

In order to find an effective and efficient countermeasure against the propagation of P2P worms, we must fully understand their propagation mechanisms. This paper presents a study on modeling the propagation of P2P worms under three different conditions. Our major contributions in this paper are firstly, we

---

* Corresponding author.
E-mail addresses: x.fan2@cqu.edu.au (X. Fan), yang@deakin.edu.au (Y. Xiang).

0167-739X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2010.04.009

Author's personal copy

propose a novel logic matrix approach to modeling the propagation of P2P worms under three different conditions; secondly, we find the impacts of two different topologies on a P2P worm's attack performance; thirdly, we find the impacts of the network-related characteristics on a P2P worm's attack performance in structured P2P networks; and fourthly, we find the impacts of two different quarantine tactics on the propagation characteristics of P2P worms in unstructured P2P networks.

The rest of the paper is organized as follows. We survey related work in Section 2. We present the proposed innovative logic matrix approach in Section 3. Then, in Section 4, we use the logic matrix approach to investigate the impacts of two different topologies on a P2P worm's attack performance, the impacts of the network-related characteristics on a P2P worm's attack performance in structured P2P networks, and the impacts of two different quarantine tactics on the propagation characteristics of P2P worms in unstructured P2P networks. Finally, Section 5 concludes this paper, and points out future research directions.

2. Related work

Mathematical models developed to model the propagation of infectious diseases have been adapted to model the propagation of worms [10]. In the area of epidemiology, both deterministic and stochastic models exist for modeling the spreading of infectious diseases [11–14]. In the network security, both deterministic and stochastic propagation models of worms based on their respective counterparts in epidemiology have emerged.

Deterministic propagation models of worms can be further divided into two categories: continuous-time and discrete-time. Since the propagation of worms is a discrete event process, discrete-time propagation models of worms are more accurate than their continuous-time counterparts in the deterministic regime.

Some typical examples of deterministic propagation models of worms are as follows:

- In the classical simple epidemic model [11–14], all hosts stay in one of only two states at any time: 'susceptible' (denoted by 'S') or 'infectious' (denoted by 'I'), and thus it is also called the SI model. Staniford et al. [15] presented a propagation model for the Code-RedI v2 worm, which is essentially the above classical simple epidemic model.
- The classical general epidemic model (Kermack–McKendrick model) [11–14] improves the classical simple epidemic model by considering removal of infectious hosts due to patching (installing software designed to fix security vulnerabilities).
- The two-factor worm model [10] extends the classical general epidemic model by accounting for removal of susceptible hosts due to patching and considering the pair-wise rate of infection as a variable rather than a constant.
- The discrete-time Analytical Active Worm Propagation (AAWP) model [16] takes into account the time an infectious host takes to infect other hosts, which is an important factor for the spread of worms [17].

Among the above models, all others are continuous-time except the last one, which is discrete-time.

Stochastic propagation models of active worms are based on the theory of stochastic processes. All of them are discrete-time in nature.

Two typical examples of stochastic propagation models of worms are as follows:

- Rohloff and Basar presented a stochastic density-dependent Markov jump process propagation model [18], for worms employing the random scanning approach, drawn from the field of epidemiology [12,19].
- Sellke et al. presented a stochastic Galton–Watson Markov branching process model [20] to characterize the propagation of worms employing the random scanning approach.

A more detailed survey on modeling the propagation process of worms can be found in our previous work [21].

The common limitation is that all of the existing models are not applicable to worms employing topological scanning. No existing model can describe the propagation of P2P worms.

Our novel logic matrix approach proposed in this paper models the propagation processes of P2P worms by difference equations of a logic matrix, which are essentially discrete-time deterministic propagation models of P2P worms. The proposed models are suitable for modeling P2P worms because these models take into account the topology of a P2P network. Existing models do not consider topology issues, which is the root cause of their common limitation mentioned above. Our work in this paper is motivated by the aspiration to invent an easy-to-employ tool to conduct network security research in general and worm propagation modeling research in particular, there being a current absence of such research instruments. Using a logic matrix in worm propagation modeling forms the major difference between this work and existing work.

In our models, temporal issues, such as the time lag for worms to infect peers and the time spent in quarantining peers, have intentionally not been considered. We acknowledge these issues and leave them as our future work. However, the paramount objective of these models is to facilitate determining the maximum number of peers in a P2P system that can be infected, which is the key element to lead effective defense mechanisms. Moreover, these temporal issues are normally strongly affected by human factors during the propagation process; these can be difficult to decide.

3. The logic matrix approach to propagation modeling of Peer-to-Peer worms

At the beginning of this section, we extend the definition of a matrix to allow its elements to be variables or constants of logic type; and term such kind of matrices logic matrices. Several operations of logic matrices are defined. Next, the topology, state, vulnerability status and quarantine status of a network are represented by its topology logic matrix, state logic matrix, vulnerability logic matrix, and quarantine logic matrix, respectively. Finally, an innovative logic matrix formulation of the propagation process of P2P worms under three different conditions is derived from first principles.

3.1. Logic matrix and its operations

We extend the definition of matrix to allow variables or constants of logic type as its elements and term such a kind of matrix a logic matrix. The values of variables of logic type can only be one of the two constants of logic type: True (denoted by 'T') or False (denoted by 'F'). If a logic matrix has only one row or one column, we can also term it a row logic vector or a column logic vector, respectively.

We define the absolute value of a variable l of logic type (denoted by $|l|$) as 1 when its value is 'T', and 0 when 'F'; and define the absolute value of a logic matrix L (denoted by $|L|$) as the total number of its elements with value 'T'. According to the above definitions, the absolute value of a logic matrix L can be worked out by summing the absolute value of each of its elements l, i.e.,

$$|L| = \sum |l|. \tag{1}$$

Author's personal copy

A logic matrix L can be inverted. The resultant $\bar{L}$ is a logic matrix of the same dimension with its elements l_{inv} being the result of the logical NOT operation of the corresponding element l of the logic matrix to be inverted. It can be defined mathematically as follows:

$$l_{\text{inv}} = \bar{l}, \tag{2}$$

where the bar over l indicates logical NOT operation.

Two logic matrices A and B can be added together if and only if their dimensions are the same, i.e., they have the same number of rows and the same number of columns. The resultant $S = A + B$ is a logic matrix of the same dimension with its element s_{ij} (in the i-th row and the j-th column) being the result of the logical OR operation of the corresponding elements a_{ij} and b_{ij} of the two logic matrices to be added together. It can be defined mathematically as follows:

$$s_{ij} = a_{ij} + b_{ij}, \tag{3}$$

where the $+$ sign between a_{ij} and b_{ij} indicates the logical OR operation.

A mutation law applies to the logic matrix addition defined above.

Two logic matrices A and B can be multiplied element-by-element if and only if their dimensions are the same, i.e., they have the same number of rows and the same number of columns. The resultant $P = AB$ is a logic matrix of the same dimension with its element p_{ij} (in the i-th row and the j-th column) being the result of the logical AND operation of the corresponding elements a_{ij} and b_{ij} of the two logic matrices to be multiplied element-by-element. It can be defined mathematically as follows:

$$p_{ij} = a_{ij}b_{ij}, \tag{4}$$

where $a_{ij}b_{ij}$ indicates the logical AND operation of a_{ij} and b_{ij}.

A mutation law applies to the logic matrix element-by-element multiplication defined above.

A logic matrix A can be multiplied by another logic matrix B in the manner of traditional matrix multiplication if and only if their inner dimensions are the same, i.e., the number of columns of the multiplicand logic matrix (the left one) is equal to the number of rows of the multiplier logic matrix (the right one). The resultant $P = AB$ is a logic matrix with the same number of rows as A and the same number of columns as B. We define the value of element p_{ij} (in the i-th row and the j-th column) of the product as determined by the following equation:

$$p_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}, \tag{5}$$

where $a_{ik}b_{kj}$ indicates the logic AND operation of a_{ik} and b_{kj}, n denotes the inner dimensions of the multiplicand and the multiplier logic matrices, and $\sum$ denotes the logical OR operation of all resultants of those logical AND operations.

Contrary to logic matrix addition and logic matrix element-by-element multiplication, a mutation law does not apply to the logic matrix multiplication in the manner of traditional matrix multiplication defined above.

Now the stage for later discussion has been set. In the next two sub-sections, we will introduce the concepts of a P2P network's topology logic matrix, state logic matrix, vulnerability logic matrix, and quarantine logic matrix, respectively; and derive our innovative logic matrix formulation of the propagation process of P2P worms under three different conditions from first principles.

3.2. The logic matrix representations

According to the traditional directed graph theory, a P2P overlay network can be represented by a directed graph G, with its set of vertices V representing all peers connected to form the network, and its set of directed edges E representing all directed links among these peers. A directed link from peer i to peer j means peer j is a neighbor of peer i, but peer i is not a neighbor of peer j if there does not exist a directed link from peer j to peer i at the same time. A peer is only able to send messages to its neighbors directly.

Topology of a P2P overlay network consisting of n peers can be represented by an n by n square matrix T with its element t_{ij} (in the i-th row and the j-th column) indicating whether there is a directed link from peer i to peer j.

In this paper, we propose a different approach from that used under the traditional directed graph theory to indicating the existence or not of a directed link. The logic constant 'T' is used to indicate there is a directed link, and the logic constant 'F' to indicate there is not. Therefore, topology of a P2P overlay network consisting of n peers can be represented by an n by n logic square matrix. We term it the topology logic matrix of the P2P overlay network.

Each row of the topology logic matrix of a P2P overlay network forms a row logic vector, which is a logic vector representation of outbound links (neighbors) of a particular peer belonging to the network. We call this row logic vector the peer's topology out-degree logic vector. Each column of the topology logic matrix of a P2P overlay network forms a column logic vector, which is a logic vector representation of inbound links of a particular peer belonging to the network. We call this logic column vector the peer's topology in-degree logic vector. For example, the i-th row of a topology logic matrix represents all outbound links (neighbors) of peer i; and the j-th column of the topology logic matrix represents all inbound links of peer j.

It can be easily derived that the values of topology in-degree and topology out-degree of each peer belonging to a P2P overlay network equate to the absolute values of the peer's topology in-degree logic vector and topology out-degree logic vector, respectively, which can be worked out by using (1).

Similarly, we represent states of all the n peers belonging to the P2P overlay network by a row logic vector S of length n with its element s_j (the j-th element) indicating whether peer j has been infected by the worm and become infectious. The logic constant 'T' is used to indicate a peer has been infected and become infectious, and the logic constant 'F' to indicate it has not. We term the above row logic vector the P2P overlay network's state logic vector. It can be derived that the total number of infected and infectious peers in a P2P overlay network equates to the absolute value of the network's state logic vector, which can be worked out by using (1).

A healthy peer vulnerable to the worm can be infected by the worm and become infectious. However, a peer which is not vulnerable to the worm will not be infected by the worm and become infectious. We represent the vulnerability status of all the n peers in the P2P overlay network by a row logic vector V of length n with its element v_j (the j-th element) indicating whether peer j is vulnerable to the worm. The logic constant 'T' is used to indicate a peer is vulnerable to the worm, and the logic constant 'F' to indicate it is not. We term the above logic row vector the P2P overlay network's vulnerability logic vector. It can be derived that the total number of peers vulnerable to the worm in a P2P overlay network equates to the absolute value of the network's vulnerability logic vector by using (1).

We represent the quarantine status of all the n peers belonging to the P2P overlay network by a row logic vector Q of length n with its element qj (the j-th element) indicating whether peer j has been

Author's personal copy

quarantined for the worm. A quarantined healthy peer will not be infected by the worm; and a quarantined infected and infectious peer will be cured and will not be infected again by the worm. The logic constant 'T' is used to indicate a peer has been quarantined, and the logic constant 'F' to indicate it has not. We term the above row logic vector the P2P overlay network's quarantine logic vector. It can be derived that the total number of quarantined peers in a P2P overlay network equates to the absolute value of the network's quarantine logic vector, which can be worked out by using (1).

3.3. The logic matrix formulation

Based on the above extensions to the matrix and its operations, and extensions to the matrix representation of a network in the traditional directed graph theory, we are now ready to derive our innovative logic matrix formulation of the propagation process of P2P worms. The derivation is based on the following assumptions:

An infected and infectious peer will send the worm packets to all other peers belonging to the same P2P overlay network to which it has a outbound link, regardless of the state (infected by the worm and infectious or not) and the quarantine status (quarantined for the worm or not) of those peers. A healthy (not infected by the worm and not infectious) peer will be infected by the worm and become infectious once it receives the worm packets from an infectious peer, provided the healthy peer is not quarantined for the worm. An infected and infectious peer will remain in that state once it receives the worm packets from an infectious peer, provided the infected and infectious peer is not quarantined for the worm. A healthy peer quarantined for the worm will not be infected by the worm; and an infected and infectious peer quarantined for the worm will be cured and will not be infected again by the worm.

The time lags from sending the worm packets, to receiving the worm packets, to having the recipient peers infected by the worm, to the peers infected by the worm becoming infectious will not be considered, nor will the time spent in quarantining peers.

There are a total of n peers belonging to a logical (not physical) P2P overlay network under consideration. Initially, there are a total of I_0 peers which are infected by the worm and infectious.

According to the above assumptions, the logical P2P overlay network's initial state (State 0) can be represented by its initial state logic vector S_0 of length n; and the absolute value of S_0 equates to the total number of peers which are initially infected by the worm and infectious (I_0), i.e.,

$$|S_0| = I_0. \tag{6}$$

Generally, state g of the logical P2P overlay network can be represented by its state logic vector S_g of length n; and the absolute value of S_g equates to the total number of peers which are infected by the worm and infectious at that state (I_g), i.e.,

$$|S_g| = I_g. \tag{7}$$

The next state (State $g+1$) of the logical P2P overlay network can be represented by its state logic vector S_{g+1} of length n; and the absolute value of S_{g+1} equates to the total number of peers which are infected by the worm and infectious at that state (I_{g+1}), i.e.,

$$|S_{g+1}| = I_{g+1}. \tag{8}$$

We notice that the logical P2P overlay network's next state represented by its state logic vector S_{g+1} is fully determined by the network's current state represented by its state logic vector S_g, the network's topology represented by its topology logic matrix T, the network's vulnerability status represented by its vulnerability logic vector V, and the network's quarantine status represented by its quarantine logic vector Q.

If all peers are vulnerable to the P2P worm, we find the relationship among S_{g+1}, S_g, T, V, and Q can be described mathematically as follows:

$$S_{g+1} = S_g + S_g T\overline{Q}. \tag{9}$$

Let S_g^{new} stand for the second term in the above equation (after the $+$ sign), the above equation can now be simplified to

$$S_{g+1} = S_g + S_g^{\text{new}}. \tag{10}$$

The term represented by S_g^{new} actually says if at State g at least one peer among those peers from which peer j has inbound links is infectious, peer j will be infected by the worm and become infectious at State $g+1$ provided peer j is not quarantined.

Since both S_g and Q are row logic vectors of length n and T is an n by n square logic matrix, S_g^{new} will be a row logic vector of length n. It can be derived that S_g^{new} is a logic vector representation of all those peers that can be infected by the worm at State $g+1$, given the network's state at State g represented by its state logic vector S_g, the network's topology represented by its topology logic matrix T, and the network's quarantine status represented by its quarantine logic vector Q. S_g^{new} may or may not include peer or peers infected by the worm at states prior to State $g+1$. Then, (9) and (10) can be easily derived.

If quarantine is not enforced at all and not all peers are vulnerable to the P2P worm, (9) will be changed to

$$S_{g+1} = S_g + S_g TV. \tag{11}$$

The term represented by the second term in the above equation (after the $+$ sign) actually says if at State g at least one peer among those peers from which peer j has inbound links is infectious, peer j will be infected by the worm and become infectious at State $g+1$ provided peer j is vulnerable to the worm.

If quarantined is not enforced at all and all peers are vulnerable to the P2P worm, (11) will be simplified to

$$S_{g+1} = S_g + S_g T. \tag{12}$$

Eq. (12) is also a special case of (9) when Q is a row logic vector with all its elements being 'F'.

Eqs. (9), (11) and (12) are actually discrete-time deterministic propagation models of P2P worms under three different condition, respectively, written in the form of difference equations of the logic matrix.

Starting from some certain state, there will be no newly infected peer to occur and thus actually, the propagation will stop. The state from which the propagation will cease is the earliest state whose state logic vector S_G satisfies the following equation:

$$|S_{G+1}| = |S_G|, \tag{13}$$

where S_{G+1} stands for the state logic vector of the state immediately after the state with state logic vector S_G.

We call the earliest state whose state logic matrix S_G satisfies (13) the final state of the P2P overlay network.

The proposed logic matrix approach essentially translates the propagation processes of P2P worms into a sequence of logic matrix operations.

4. Simulation experiments: applications of the logic matrix approach

Our evaluation metric for attack performance in this paper is a P2P worm's coverage rate (denoted by c) in a logical P2P overlay network. It is defined as the ratio of number of peers in the network that can be infected by the worm to number of peers in the network

Author's personal copy

that are vulnerable to the worm. It can be worked out by using the following equation:

$$c = \frac{|S_G|}{|V|}, \tag{14}$$

where S_G is the state logic vector of the network when the propagation process has just stopped, and V is the vulnerability logic vector to the worm of the network.

One of our evaluation metrics for network-related characteristics of P2P networks in this paper is vulnerability rate (denoted by v) to a P2P worm of a logical P2P overlay network, which is defined as the ratio of number of peers in the network that are vulnerable to the worm to total number of peers in the network. It can be worked out by using the following equation:

$$v = \frac{|V|}{n}, \tag{15}$$

where V is the vulnerability logic vector to the worm of the network, and n is total number of peers in the network.

The other two of our evaluation metrics for network-related characteristics of P2P networks in this paper are topology out-degree, which refers to the number of logical neighbors maintained by each peer locally; and network size, which refers to the total number of peers in a P2P network.

Our defense-related evaluation metric in this paper is quarantine rate (denoted by q) for a P2P worm of a logical P2P overlay network. It is defined as the ratio of number of peers belonging to the network that are quarantined for the worm to total number of peers belonging to the network; and can be worked out by using the following equation:

$$q = \frac{|Q|}{n}, \tag{16}$$

where Q is the quarantine logic vector for the worm of the network and n is total number of peers belonging to the network.

We apply the proposed logic matrix approach in our simulation experiments under the following three different conditions using MathWorks' MATLAB.

4.1. All peers being vulnerable to the P2P worm and no quarantine at all

In this case, we investigate the impacts of the two different topologies, namely the simple random graph topology and the pseudo power law topology on the coverage rate of P2P worms.

4.1.1. The simple random graph topology

We investigate the impacts of the two parameters, namely the number of initially infected computers belonging to a P2P network and the mean value of topology out-degree of the network, on the coverage rate of P2P worms in the network.

Our implementation in MATLAB assumes there are a total of 10,000 peers (computers) belonging to the logical P2P overlay network under consideration. Therefore, the topology of the overlay network is represented by its topology logic matrix, which is a 10,000 by 10,000 square logic matrix; and its initial state is represented by its initial state logic vector, which is a 1 by 10,000 logic matrix (row logic vector). In the experiments conducted for this sub-section, we assume each peer has the same value of topology out-degree. Peers to which each peer has outbound links are randomly selected from all peers except the peer itself belonging to the overlay network, which means we do not allow loop, that is, no peer has an outbound link to itself. Therefore, we call the topology of the overlay network in the experiments conducted for this sub-section the simple random graph topology.

Table 1
A list of the experimental results when there is only 1 initially infected peer, which is randomly selected from all peers.

Mean value of topology out-degree	Mean value of coverage rate (%)	Coefficient of variation of coverage rate (%)
1	1.23	54.81
2	79.64	0.68
3	94.08	0.27
4	98.06	0.16
5	99.31	0.09

We conduct our experiments with MATLAB under different combinations of values of the number of initially infected computers and the mean value of topology out-degree.

Firstly, we fix the number of initially infected peers (computers) belonging to the overlay network to be 1, and try to find out the impact of mean value of topology out-degree on the coverage rate of P2P worms in the overlay network. The initially infected peer is randomly select from all peers belonging to the overlay network. A total of 5 scenarios listed in Table 1 are investigated. The experiment for each scenario is repeated 100 times. Next, the mean value of coverage rate and coefficient of variation of coverage rate are worked out. Results from the experiments are listed in Table 1.

As shown by Table 1, the mean value of topology out-degree has great impact on both the mean value and coefficient of variation of coverage rate of P2P worms in the overlay network featuring the simple random graph topology. An increase in the mean value of topology out-degree results in an increase in the mean value of coverage rate but a decrease in the coefficient of variation of coverage rate. When the mean value of topology out-degree is increased to 3, the mean value of coverage rate is increased to over 90% and its coefficient of variation becomes very small, which indicates 3 is the minimum mean value of topology out-degree which can make a P2P worm able to infect most peers with very high certainty.

Next, we fix the number of initially infected peers (computers) belonging to the overlay network to be 10/100, and repeat the above experiments. Results from the experiments are listed in Table 2.

Table 2 shows similar trends to those shown by Table 1, which indicates the impact of number of initially infected peers on the coverage rate of a P2P worm in the overlay network featuring the simple random graph topology is insignificant.

4.1.2. The pseudo power law topology

We investigate the impacts of the two parameters, namely the number of initially infected computers belonging to a P2P network and the maximum value of topology out-degree of the network, on the coverage rate of P2P worms in the network.

In the experiments conducted for this sub-section, we assume only a very small number (10 in our experiments) of peers have the maximum value of topology out-degree, and all other peers have the minimum value (1 in our experiments) of topology out-degree. Although the distribution of topology out-degree in our experiments does not strictly follow a power law, it does have the most important features of power law distribution, namely peers with the maximum value of topology out-degree are rare and most peers have the minimum value of topology out-degree. Therefore, we call the topology of the overlay network in the experiments conducted for this sub-section the pseudo power law topology.

We conduct our simulation under different combinations of values of the number of initially infected computers and the maximum value of topology out-degree.

Firstly, we fix the number of initially infected peers (computers) belonging to the overlay network to be 1, and try to find out the impact of the maximum value of topology out-degree on the

Author's personal copy

Table 2
A list of the experimental results when there are a total of 10/100 initially infected peers, all of which are randomly selected from all peers.

Mean value of topology out-degree	Mean value of coverage rate (%)		Coefficient of variation of coverage rate (%)	
	Initially infected peers = 10	Initially infected peers = 100	Initially infected peers = 10	Initially infected peers = 100
1	4.28	13.53	16.22	5.43
2	79.80	80.06	0.63	0.62
3	94.10	94.16	0.27	0.26
4	98.03	98.06	0.15	0.15
5	99.30	99.31	0.08	0.09

Table 3
A list of the experimental results when there is only 1 initially infected peer, which is randomly selected from all peers.

Maximum value of topology out-degree	Mean value of coverage rate (%)	Coefficient of variation of coverage rate (%)
100	3.17	200.74
1000	13.83	209.20
2000	14.54	226.10

Table 4
A list of the experimental results when there are a total of 10 initially infected peers, all of which are randomly selected from all peers.

Maximum value of topology out-degree	Mean value of coverage rate (%)	Coefficient of variation of coverage rate (%)
100	11.25	79.51
1000	33.06	111.27
2000	36.23	120.07

coverage rate in the overlay network. The initially infected peer is randomly select from all peers belonging to the overlay network. A total of 5 scenarios are investigated. In the experiments conducted for this sub-section, we assume each peer has either the maximum value of topology out-degree or the minimum value of topology out-degree. Peers to which each peer has outbound links are randomly selected from all peers except the peer itself belonging to the overlay network. The experiment for each scenario is repeated 100 times. Next, the mean value of coverage rate and coefficient of variation of coverage rate are worked out. Results from the experiments are listed in Table 3.

As shown by Table 3, when all initially infected peers are randomly selected from all peers, the maximum value of topology out-degree has a little impact on both the mean value and coefficient of variation of coverage rate of P2P worms in the overlay network featuring the pseudo power law topology. An increase in the maximum value of topology out-degree results in a little increase in the mean value of coverage rate and a little increase in coefficient of variation of coverage rate as well, which indicates the small gain in coverage rate could be offset by the small loss in certainty. The worm is not able to infect most peers with high certainty.

Then, we fix the number of initially infected peers (computers) belonging to the overlay network to be 10, and repeat the above experiments. Results from the experiments are listed in Table 4.

Table 4 shows similar trends (just an insignificantly higher coverage rate and an insignificantly lower coefficient of variation of coverage rate) to those shown by Table 3, which indicates, when all initially infected peers are randomly selected from all peers, the impact of number of initially infected peers on the coverage rate of a P2P worm in the overlay network featuring the pseudo power law topology is insignificant.

Finally, initially infected peers are randomly select from only those peers with maximum topology out-degree and we repeat all of the above experiments described in this sub-section. Results from the experiments are listed in Tables 5 and 6.

Table 5
A list of the experimental results when there is only 1 initially infected peer, which is randomly selected from only those peers with maximum topology out-degree.

Maximum value of topology out-degree	Mean value of coverage rate (%)	Coefficient of variation of coverage rate (%)
100	20.74	26.65
1000	78.21	11.17
2000	95.33	0.89

Table 6
A list of the experimental results when there are a total of 10 initially infected peers, all of which are randomly selected from only those peers with maximum topology out-degree.

Maximum value of topology out-degree	Mean value of coverage rate (%)	Coefficient of variation of coverage rate (%)
100	38.50	1.53
1000	85.19	0.41
2000	95.94	0.19

As shown by Tables 5 and 6, when all initially infected peers are randomly selected from only those peers with maximum topology out-degree, the maximum value of topology out-degree has a great impact on both the mean value and coefficient of variation of coverage rate of P2P worms in the overlay network featuring the pseudo power law topology. An increase in the maximum value of topology out-degree results in an increase in the mean value of coverage rate but a decrease in the coefficient of variation of coverage rate. However, the impact of the number of initially infected peers is insignificant. When the maximum value of topology out-degree reaches 2000, the worm is able to infect most peers with very high certainty, regardless of the number of initially infected peers.

4.2. Not all peers being vulnerable to the P2P worm and no quarantine at all

In a structured P2P network, the topology out-degree d of each peer is a constant. It is characterized by the following probability distribution:

$$\begin{cases} P(d = k) = 1 \\ P(d \neq k) = 0, \end{cases} \tag{17}$$

where k is a constant.

In this sub-section, we only consider structured P2P networks. Therefore, all peers in the network have the same topology out-degree.

Our objective is to investigate the impacts of the network-related characteristics (measured by the evaluation metrics: vulnerability rate v, topology out-degree d, and network size n) on a P2P worm's attack performance in structured P2P networks (measured by the evaluation metric: coverage rate c).

Our simulation experiments include scenarios with vulnerability rate of 1.0. Experimental results from them set the benchmark to compare to. When all peers are vulnerable to the worm, (12) instead of (11) forms the foundation of our implementation of the proposed logic matrix approach. Otherwise, our implementation is based on (11).

Author's personal copy

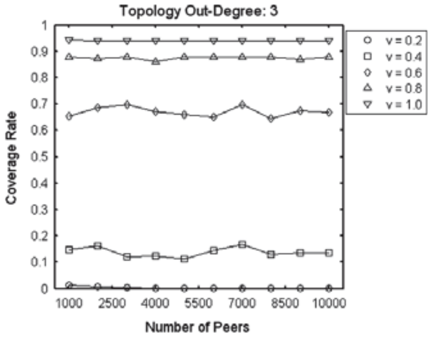


Fig. 1. Coverage rate as a function of vulnerability rate and network size when topology out-degree is fixed at 3.

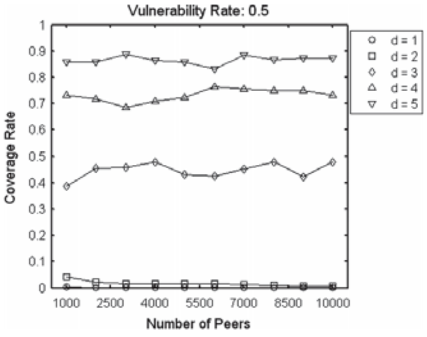


Fig. 2. Coverage rate as a function of topology out-degree and network size when vulnerability is fixed at 0.5.

Our simulation experiments are based on the following assumptions:

- Topology out-degree (d) of each peer in the structured P2P network under consideration strictly follows the probability distribution (17). Neighbors of a peer are randomly selected from all other peers except the peer itself.
- Peers vulnerable to the worm are selected randomly from all peers in the network.
- There is only 1 initially infected peer, which is selected randomly from all peers in the network that are vulnerable to the worm.

Based on the above assumptions, we first populate the topology logic matrix of the structured P2P network under consideration by letting the probability that a randomly selected peer has k neighbors follow (17). Then, we populate the vulnerability logic vector of the network, before populating the initial state logic vector of the network.

We conduct our simulation experiments for the three different sets of scenarios. Each of our simulation experiment is repeated 100 times, and then average values of coverage rate are reported as final results.

For our first set of scenarios, we fix topology out-degree at 3. We let vulnerability rate vary from 1.0 to 0.2 with step size −0.2; and let network size vary from 1000 to 10,000 with step size 1000. A vulnerability rate of 1.0 actually means all peers in the network are vulnerable to the worm. The experimental results from the above set of scenarios are illustrated by Fig. 1.

Fig. 1 reveals that under the set conditions, the coverage rate of a P2P worm in a logical P2P overlay network will decrease if vulnerability rate is decreased. This is sensible since more vulnerable peers in the network naturally lead to higher attack performance measured by the coverage rate. The upper bound of the coverage rate is approximately 0.95. It is achieved when all peers are vulnerable, i.e., $v = 1.0$ (the top curve in Fig. 1). The lower bound of the coverage rate is close to 0. It is achieved when 20% peers are vulnerable, i.e., $v = 0.2$ (the bottom curve in Fig. 1). The coverage rate drops significantly from above 0.6 to below 0.2 when vulnerability rate is decreased from 0.6 to 0.4. The above findings imply that both attackers and defenders can manipulate vulnerability rate v to improve or worsen attack performance, respectively, and more importantly that limiting vulnerability rate to be below 0.4 is critical to defenders.

Fig. 1 also shows that, when network size is in the range 1000–10,000 inclusive, it has no significant impact on attack performance measured by the coverage rate if both topology out-degree and vulnerability rate are fixed. This finding implies that we can choose a smaller value in the range 1000–10,000 for network size n in our later experiments to shorten simulation time, and that neither attackers nor defenders can manipulate network size n to improve or worsen attack performance, respectively.

For our second set of scenarios, we fix vulnerability rate at 0.5. We let topology out-degree vary from 1 to 5 with step size 1; and let network size vary from 1000 to 10,000 with step size 1000. The experimental results from the above set of scenarios are illustrated by Fig. 2.

Fig. 2 reveals that under the set conditions, the coverage rate of a P2P worm in a logical P2P overlay network will increase if topology out-degree is increased. This is sensible since the more neighbors a peer in the network has naturally leads to higher attack performance measured by the coverage rate. The upper bound of the coverage rate is approximately 0.85. It is achieved when all peers have 5 neighbors, i.e., $d = 5$ (the top curve in Fig. 2). The lower bound of the coverage rate is 0. It is achieved when all peers have only 1 neighbor, i.e., $d = 1$ (the bottom curve in Fig. 1). The coverage rate drops significantly from above approximately 0.4 to below 0.05 when topology out-degree decreased from 3 to 2. The above findings imply that both attackers and defenders can manipulate topology out-degree d to improve or worsen attack performance, respectively, and more importantly that limiting topology out-degree to be below or equal to 2 is critical to defenders.

Based on the common finding from our first 2 sets of simulation experiments that when network size is in the range 1000–10,000 inclusive, it has no significant impact on attack performance, for our third set of scenarios, we fix network size at 5000 to shorten simulation time. We investigate the two cases given below:

Case 1—In this case, we let vulnerability rate vary from 0.1 to 1.0 with step size 0.1; and let topology out-degree vary from 1 to 5 with step size 1. Here, our focus is on the impact of vulnerability rate rather than topology out-degree on attack performance measured by the coverage rate. Therefore, we choose a smaller step size for vulnerability rate, but only a few topology out-degree values are investigated.

The experimental results from the above case are illustrated by Fig. 3. Fig. 3 reveals that generally the coverage rate of a P2P worm in a logical P2P overlay network will increase if vulnerability rate is increased. This is sensible since more vulnerable peers in the network naturally lead to higher attack performance measured by the coverage rate.

More importantly, Fig. 3 also shows that the takeoff points on the curves do not correspond to the same value of vulnerability rate. Here, takeoff point refers to the point on a curve in Fig. 3 immediately to the right of which the slope of the curve increases

Author's personal copy

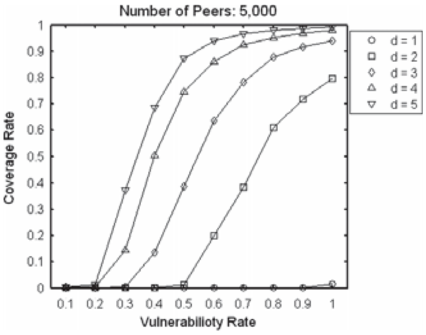


Fig. 3. Coverage rate as a function of topology out-degree and vulnerability rate when network size is fixed at 5000 (Case 1).

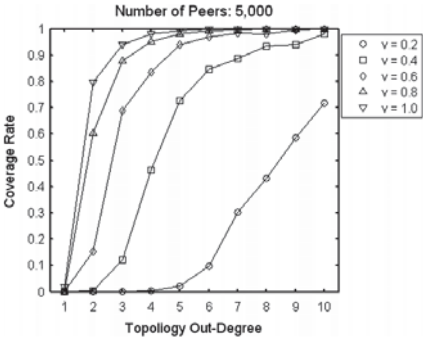


Fig. 4. Coverage rate as a function of vulnerability rate and topology out-degree when network size is fixed at 5000 (Case 2).

dramatically. For instance, when topology out-degree is fixed at 5, the takeoff point corresponds to vulnerability rate 0.2; and when topology out-degree is reduced to 3, the takeoff point corresponds to vulnerability rate 0.3. Generally, the corresponding vulnerability rate will increase if topology out-degree is reduced. This is sensible since fewer neighbors demand more vulnerable peers to achieve the same attack performance. It can be found from Fig. 3 that 0.2 is a critical value of vulnerability rate, since if vulnerability rate is below that value the worm cannot propagate successfully in the network.

Case 2—In this case, we let topology out-degree vary from 1 to 10 with step size 1; and let vulnerability rate vary from 0.2 to 1.0 with step size 0.2. Here, our focus is on the impact of topology out-degree rather than vulnerability rate on attack performance measured by the coverage rate. Therefore, a large range of topology out-degree values are investigated, but we choose a larger step size for vulnerability rate.

The experimental results from the above case are illustrated by Fig. 4. Fig. 4 reveals that generally the coverage rate of a P2P worm in a logical P2P overlay network will increase if topology out-degree is increased. This is sensible since the more neighbors a peer in the network has naturally leads to higher attack performance measured by the coverage rate.

More importantly, Fig. 4 also shows that the takeoff points on the curves do not correspond to the same value of topology out-degree. When vulnerability rate is fixed at 1.0, the takeoff point corresponds to topology out-degree 1; and when vulnerability rate is reduced to 0.2, the takeoff point corresponds to topology out-degree 5. Generally, the corresponding topology out-degree will increase if vulnerability rate is reduced. This is sensible since fewer vulnerable peers demand more neighbors a peer in the network has, to achieve the same attack performance.

4.3. All peers being vulnerable to the P2P worm and quarantine being existent

In an unstructured P2P network, the topology out-degree (d) of each peer is a variable. It is characterized by the following power law distribution:

$$\begin{cases} D_{\min} \le k \le D_{\max} \\ P(d = k) = \dfrac{C}{k^A} \\ P(d \ne k) = 0, \end{cases} \tag{18}$$

where $D_{\min}$ and $D_{\max}$ stands for the minimum topology out-degree and maximum topology out-degree, respectively, A represents the power law degree, and C is a constant. The set of equations (18) gives the probability that a randomly selected peer has k neighbors.

In this subsection, we only consider unstructured P2P networks. Therefore, not all peers in the network have the same topology out-degree.

Our paramount objective is to find a quarantine tactic whose enforcement will lead to a lower attack performance (measured by the attach-related evaluation metric: coverage rate c) at a lower cost of defense effort (measured by the defense-related evaluation metric: quarantine rate q).

According to probability theory, the following equations must hold:

$$1 = \sum_{k=D_{\min}}^{D_{\max}} P(d = k) = C \sum_{k=D_{\min}}^{D_{\max}} \frac{1}{k^A}, \tag{19}$$

$$E(d) = \sum_{k=D_{\min}}^{D_{\max}} kP(d = k) = C \sum_{k=D_{\min}}^{D_{\max}} \frac{1}{k^{A-1}}, \tag{20}$$

where $E(d)$ stands for expected value of topology out-degree.

Then, it can be easily derived from (19) and (20) that the power law degree A is a function of $D_{\min}$, $D_{\max}$, and $E(d)$ described implicitly by the following equation:

$$E(d) = \frac{\sum\limits_{k=D_{\min}}^{D_{\max}} \frac{1}{k^{A-1}}}{\sum\limits_{k=D_{\min}}^{D_{\max}} \frac{1}{k^A}}. \tag{21}$$

Finally, once the power law degree A is determined according to (20), given $D_{\min}$, $D_{\max}$, and $E(d)$, the constant C can be worked out according to (19) or (20).

The most important feature of the above power law distribution of topology out-degree in the unstructured P2P system is that there are fewer peers with larger topology out-degree than those with smaller topology out-degree.

Let $D_{\min} = 1$, $D_{\max}$ vary from 100 to 1000 and the expected value of topology out-degree $E(d)$ vary from 2 to 32, we numerically determine power law degree A. The results are shown in Fig. 5.

Fig. 5 shows that a larger maximum topology out-degree requires a larger power law degree, and that a larger expected value of topology out-degree demands a smaller power law degree.

Author's personal copy

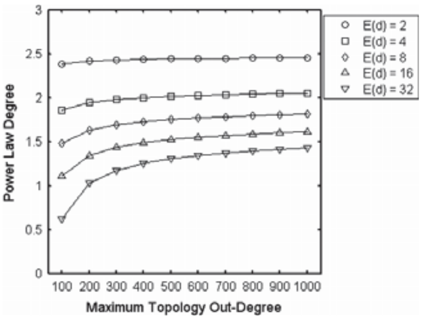


Fig. 5. Power law degree as a function of the maximum topology out-degree and expected value of topology out-degree given the minimum topology out-degree being 1.

Our simulation experiments are based on the following assumptions:

- Topology out-degree (d) of each peer belonging to the unstructured P2P network under consideration strictly follows the power law distribution (18). $E(d) = 3$, $D_{\min} = 1$, and $D_{\max}$ varies from 100 to 1000 with step size 100. Neighbors of a peer are randomly selected from all other peers except the peer itself.
- Peers quarantined are selected accordingly, based on the quarantine tactics enforced, which are detailed in the next two subsections.
- There is only 1 initially infected peer, which is selected randomly from all peers not quarantined.
- We conduct our simulation experiments for two different values of n (total number of peers belonging to the system). We first assume n to be 5000 and then double it, i.e., assume n to be 10,000. We believe 10,000 peers are sufficient for our simulation experiments, and intend to investigate whether 5000 peers will generate significantly different results.

Based on the above assumptions, we populate the topology logic matrix of the unstructured P2P network under consideration by letting the probability that a randomly selected peer has k neighbors follow (18). How to populate the quarantine logic vector of the network is detailed later. Once it is populated, we can populate the initial state logic vector of the network.

Our simulation experiments include scenarios with no quarantine at all. The experimental results from these set the benchmark to compare to. When there is no quarantine, (12) instead of (9) forms the foundation of our implementation of the proposed logic matrix approach. When quarantine is enforced, our implementation is based on (9).

We conduct our simulation experiments for two different quarantine tactics, namely random quarantine and larger topology out-degree priority quarantine. Each of our simulation experiment is repeated 100 times, and then the average values of coverage rate are reported as final results.

4.3.1. Random quarantine

Random quarantine means peers quarantined are randomly selected from all peers. We populate the quarantine logic vector of the unstructured P2P network under consideration by letting each peer have the same probability of being quarantined when this quarantine tactic is enforced. Then, we populate the initial state logic vector of the network.

We conduct our experiments for the 5 sets of scenarios with quarantine rate q varying from 0 to 0.4 with step size 0.1. A quarantine rate of 0 actually means no quarantine at all. We include no quarantine as a special case of random quarantine, which facilitates comparison of experimental results.

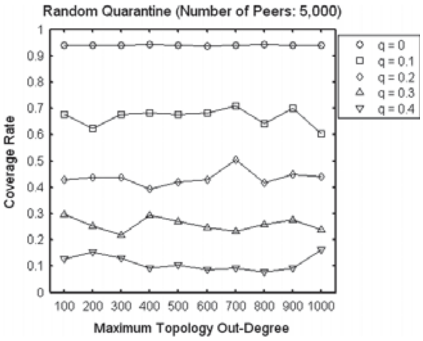


Fig. 6. Coverage rate under random quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 5000 peers in the P2P system.

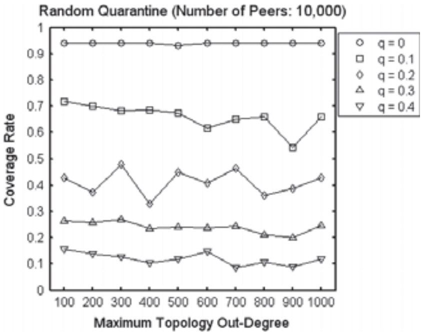


Fig. 7. Coverage rate under random quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 10,000 peers in the P2P system.

The experimental results from random quarantine are illustrated by Figs. 6 and 7 for the two cases: n (total number of peers belonging to the P2P network) = 5000 and $n = 10{,}000$, respectively.

Figs. 6 and 7 reveal that generally, coverage rate of a P2P worm in a logical P2P overlay network will decrease if quarantine rate is increased. This is sensible because a higher defense effort will naturally lead to a lower attack performance. However, as mentioned previously, our paramount objective is to find a quarantine tactic whose enforcement will lead to a lower attack performance at a lower cost of defense effort. Therefore, the above finding cannot serve our paramount objective.

Figs. 6 and 7 also show that maximum topology out-degree has no significant impact on attack performance and defense effort when it is in the range 100–1000 inclusive, and that 5000 peers will not generate significantly different results. The implications of the above findings include that we can choose the smallest value of $D_{\max}$ (100) and the smaller value of n (5000) in our future experiments to shorten simulation time, and that neither attackers nor defenders can manipulate n or $D_{\max}$ to improve attack performance or reduce defense effort, respectively.

Author's personal copy

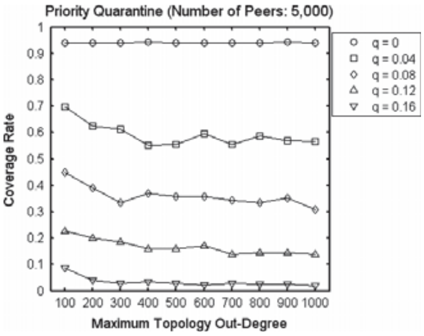


Fig. 8. Coverage rate under priority quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 5000 peers in the P2P system.

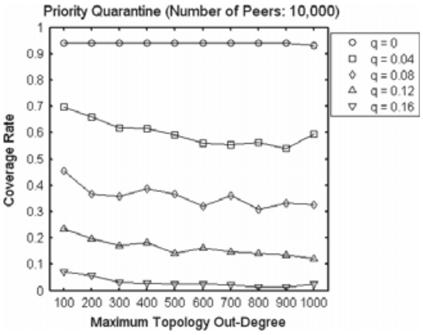


Fig. 9. Coverage rate under priority quarantine as a function of maximum topology out-degree and quarantine rate when there are a total of 10,000 peers in the P2P system.

4.3.2. *Larger topology out-degree priority quarantine*

Larger topology out-degree priority quarantine means peers with larger topology out-degree are quarantined prior to peers with smaller topology out-degree.

When this quarantine tactic is enforced, we populate the quarantine logic vector of the unstructured P2P network under consideration by following the procedure given below:

Firstly, we work out the absolute value of each peer's topology out-degree logic vector. Secondly, all peers are sorted in descending order of the absolute value calculated above. By doing this, we actually sort all peers into a list in descending order of number of neighbors since, as mentioned previously, each peer's topology out-degree logic vector is a logic vector representation of its outbound links (neighbors). Thirdly, we quarantine peers in the same order as their order in the sorted list of peers. Then, we populate the initial state logic vector of the network.

We conduct our experiments for the 5 sets of scenarios with quarantine rate q varying from 0 to 0.16 (40% of 0.4, which is the maximum quarantine rate investigated under random quarantine) with step size 0.04.

The experimental results from larger topology out-degree priority quarantine are illustrated by Figs. 8 and 9 for the two cases: n (total number of peers belonging to the P2P network) $= 5000$ and $n = 10{,}000$, respectively.

Figs. 8 and 9 reveal that generally, coverage rate of a P2P worm in a logical P2P overlay network will decrease if quarantine rate is increased. Figs. 8 and 9 also show that maximum topology out-degree has no significant impact on attack performance and defense effort when it is in the range 100–1000 inclusive, and that 5000 peers will not generate significantly different results. The above findings are the same as those from random quarantine.

If we compare the bottom curve in Fig. 6 to the bottom curve in Fig. 8, it can be found that larger topology out-degree priority quarantine demands a lower defense effort (quarantine rate $q = 0.16$) to achieve a lower attack performance (coverage rate $c < 0.1$), and that random quarantine demands a higher defense effort (quarantine rate $q = 0.4$) to achieve a higher attack performance (coverage rate $c < 0.2$). The same result as above can be found if we compare the bottom curve in Fig. 7 to the bottom curve in Fig. 9. The above finding exactly serves our paramount objective, which is to find a quarantine tactic whose enforcement will lead to a lower attack performance at a lower cost of defense effort.

Therefore, according to our experimental results, larger topology out-degree priority quarantine outperforms random quarantine. Larger topology out-degree priority quarantine is exactly the quarantine tactic we are looking for since it demands only 40% (0.16/0.4) defense effort to achieve 50% (0.1/0.2) attack performance, compared to random quarantine. In other words, larger topology out-degree priority quarantine is much more efficient than random quarantine.

5. Conclusion

This paper presents a study on modeling the propagation processes of P2P worms. In this paper, based on our definitions of logic matrix and its operations, we have proposed the logic matrix representation of a P2P overlay network's topology, topology out-degree, topology in-degree, state, vulnerability status, and quarantine status; and derived our unique logic matrix approach to modeling the propagation of P2P worms. Based on this model, we find the impacts of the two different topologies on a P2P worm's attack performance, the impacts of the network-related characteristics on a P2P worm's attack performance in structured P2P networks, and the impacts of the two different quarantine tactics on the propagation characteristics of P2P worms in unstructured P2P networks.

To the best of our knowledge, we are the first using logic matrix in network security research in general and worm propagation research in particular. The proposed approach's ease of employment makes it an attractive instrument to conduct worm propagation research. We have demonstrated the innovative logic matrix formulation proposed in this paper, which are discrete time deterministic propagation models of P2P worms described by difference equations of logic matrix, is a highly effective and efficient tool for investigating the propagation processes of P2P worms.

In the future, we plan to extend P2P worm models by considering temporal issues, such as the time lag for worms to infect peers and the time spent in quarantining peers. Time series-based matrices can be potentially used in the extended model. We will also look for a more effective and efficient quarantine tactic if temporal issues are considered.

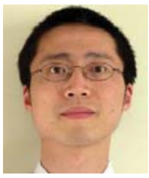
To make it more practical to accommodate the dynamic P2P network where peers can join and leave a network, a P2P network's topology logic matrix needs to include peers that will join the network in the future. It also needs to be updated once a peer joins or leaves the network. This could be simulated by randomly selecting peers joining and leaving the network, which means the topology logic matrix of the P2P network is constantly changing. We are going to incorporate the above idea into the simulation conducted previously.

Author's personal copy

References

[1] N. Weaver, V. Paxson, S. Staniford, R. Cunningham, A taxonomy of computer worms, in: WORM'03, Washington, DC, USA, 2003, pp. 11–18.
[2] D. Moore, C. Shannon, J. Brown, Code-red: a case study on the spread and victims of an internet worm, in: IMW'02, Marseille, France, 2002, pp. 273–284.
[3] C.C. Zou, D. Towsley, W. Gong, On the performance of internet worm scanning strategies, University of Massachusetts Technical Report: TR-03-CSE-07, 2003.
[4] C.C. Zou, D. Towsley, W. Gong, S. Cai, Routing worm: a fast, selective attack worm based on IP address information, in: PADS'05, 2005, pp. 199–206.
[5] Z. Chen, C. Ji, Importance-scanning worm using vulnerable-host distribution, in: IEEE GLOBECOM, 2005, pp. 1779–1784.
[6] Z. Chen, C. Ji, A self-learning worm using importance scanning, in: WORM'05, Fairfax, VA, USA, 2005, pp. 22–29.
[7] E.H. Spafford, The internet worm program: an analysis, ACM SIGCOMM Computer Communication Review 19 (1989) 17–57.
[8] I. Arce, E. Levy, An analysis of the Slapper worm, IEEE Security & Privacy (2003) 82–87.
[9] W. Yu, Analyze the worm-based attack in large scale P2P networks, in: The 8th IEEE International Symposium on High Assurance Systems Engineering, HASE 2004, 2004.
[10] C.C. Zou, W. Gong, D. Towsley, Code Red worm propagation modeling and analysis, in: CCS'02, Washington, DC, USA, 2002, pp. 138–147.
[11] R.M. Anderson, R.M. May, Infectious Diseases of Humans: Dynamics and Control, Oxford University Press, Oxford, 1991.
[12] H. Andersson, T. Britton, Stochastic Epidemic Models and their Statistical Analysis, Springer-Verlag, New York, 2000.
[13] N.T. Bailey, The Mathematical Theory of Infectious Diseases and its Applications, Hafner Press, New York, 1975.
[14] J.C. Frauenthal, Mathematical Modeling in Epidemiology, Springer-Verlag, New York, 1980.
[15] S. Staniford, V. Paxson, N. Weaver, How to own the internet in your spare time, in: Security'02, San Francisco, CA, USA, 2002, pp. 149–167.
[16] Z. Chen, L. Gao, K. Kwiat, Modeling the spread of active worms, in: IEEE INFOCOM, 2003, pp. 1890–1900.
[17] Y. Wang, C. Wang, Modeling the effects of timing parameters on virus propagation, in: WORM'03, Washington, DC, USA, 2003, pp. 61–66.
[18] K. Rohloff, T. Basar, Stochastic behavior of random constant scanning worms, in: 14th ICCCN, San Diego, CA, USA, 2005, pp. 339–344.
[19] D.J. Daley, J. Gani, Epidemic Modelling: An Introduction, Cambridge University Press, Cambridge, 1999.
[20] S. Sellke, N.B. Shroff, S. Bagchi, Modeling and automated containment of worms, in: DSN'05, 2005, pp. 528–537.
[21] Y. Xiang, X. Fan, W. Zhu, Propagation of active worms: a survey, International Journal of Computer Systems Science & Engineering 24 (2009) 157–172.

Xiang Fan is currently with School of Management and Information Systems, Central Queensland University. His research interests include network security in general and propagation of active worms in particular. He is currently working in a research group developing active defense systems against large-scale network attacks and new Internet security countermeasures.



Dr. Yang Xiang is currently with School of Information Technology, Deakin University, Australia. His research interests include network and system security, and wireless systems. In particular, he is currently working in a research group developing active defense systems against large-scale network attacks and new Internet security countermeasures. He has served as PC Chair for the 2009 3rd International Conference on Network and System Security (NSS 2009), the 11th IEEE International Conference on High Performance Computing and Communications (HPCC 2009), and the 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2008). He has been PC member for many international conferences such as IEEE ICC, IEEE GLOBECOM and IEEE ICPADS. He is on the editorial board of Journal of Network and Computer Applications.

Modeling and Simulating the Propagation of Structured Peer-to-Peer Worms

Xiang Fan, William W. Guo, and Mark Looi
School of Engineering and Technology, Central Queensland University, Rockhampton, Australia
Email: {x.fan2, w.guo, m.looi}@cqu.edu.au

***Abstract*— Peer-to-Peer (P2P) worms have become the most serious problem in the Internet because of its adaptive propagation features. Due to the complexity of the problem, no existing work has solved the problem of modeling the propagation of P2P worms, especially when quarantine of peers is enforced. This paper presents a study on modeling and simulating the propagation of structured P2P worms under random quarantine. Based on the extended logic matrix approach for modeling the propagation of P2P worms, we simulated the impacts of topology out-degree, the number of initially infected peers, and quarantine rate on the propagation characteristics of P2P worms in a structured logical P2P overlay network. The optimal quarantine ratios leading to maximum quarantine efficiency with topology out-degree under 3 have been revealed by our simulation. This approach is easy to use, which further enhances its potential of wide adoption in securing peer-to-peer networks in the future.**

***Index Terms*— Modeling, Simulating, Propagation, Structured, Peer-to-Peer, Worms**

I. INTRODUCTION

Peer-to-peer networks have become popular in networked communications but are often vulnerable to viruses spreading. Great effort has been made on securing applications of peer-to-peer networks [1-4]. Computer/Internet worms can be classified according to the techniques by which they discover new targets to infect. Scanning to probe a set of vulnerable hosts is the technique widely employed by worms [5]. Scanning can be implemented differently and different implementations lead to several different types of scanning, such as random scanning, localized scanning [6], sequential scanning [7], routable scanning [8], selective scanning [8], importance scanning [9][10], and topological scanning. Topological scanning was employed by the Morris Internet Worm of 1988 as its target discovery technique [11].

Computer/Internet worms employing all other types of scanning except topological scanning among the above types do not need to have any knowledge on topology of the network they intend to propagate across. On the contrary, computer/Internet worms employing topological scanning must have some information on the network they intend to propagate over, or have the capability to discover that information if they do not have it in advance. Therefore, computer/Internet worms employing topological scanning are also called topology-aware worms.

Typical examples of topology-aware worms are worms attacking a flaw in a Peer-to-Peer (P2P) application and propagating across the P2P network by getting lists of peers from their victims and directing their subsequent attacks to those peers. This sort of topology-aware worms is called P2P worms. The Slapper worm of 2003 was a typical example of P2P worms [12]. The subsequent appearance of variations of the Slapper worm (the Slapper.B worm a.k.a. Cinik and the Slapper.C worm a.k.a. Unlock) indicates that P2P worms are becoming increasingly complex and sophisticated [12].

Due to recent popularity of P2P systems with increasing number of users, P2P systems have become the most effective vehicles for topology-aware worms to achieve fast propagation across the Internet. Propagation of P2P worms on top of P2P systems can result in significant damages as illustrated by [13]. P2P worms are posing a serious challenge to network security.

In order to find an effective and efficient counter measure against propagation of P2P worms, we must fully understand their propagation characteristics. This paper presents a study on modeling and simulating the propagation of structured P2P worms under random quarantine. In this paper, we firstly expand our recently proposed logic matrix approach for modeling propagation of P2P worms. We then use this extended model to simulate the impacts of topology out-degree, the number of initially infected peers, and quarantine rate on propagation characteristics of P2P worms in a structured logical P2P overlay network. Based on the simulation results, optimal quarantine ratios that can lead to maximum quarantine efficiency with a topology out-degree of 3 or less are revealed.

The rest of the paper is organized as follows. We survey related work in Section 2, before presenting our proposed innovative logic matrix approach in Section 3. Then, in Section 4, we apply the proposed approach to our simulation experiments to investigate the impacts of topology out-degree, the number of initially infected peers, and quarantine rate on propagation characteristics of P2P worms in a structured logical P2P overlay network, and to look for the relationship between quarantine efficiency and quarantine ratio. Finally, Section V concludes this paper, and points out future research directions.

II. RELATED WORK

Mathematical models developed to model the propagation of infectious diseases have been adapted to model the propagation of computer/Internet worms [14]. In epidemiology area, both deterministic and stochastic models exist for modeling the spreading of infectious diseases [15-18]. In network security area, both deterministic and stochastic propagation models of computer/Internet worms based on their respective counterpart in epidemiology area have emerged.

Deterministic propagation models of computer/Internet worms can be further divided into two categories: continuous-time and discrete-time. Since the propagation of computer/Internet worms is a discrete event process, discrete-time propagation models of computer/Internet worms is more accurate than its continuous-time counterparts in the deterministic regime. Some typical examples of deterministic propagation models of computer/Internet worms are as follows.

1) In the classical simple epidemic model [15-18], all hosts stay in one of the only two states at any time: 'susceptible' (denoted by 'S') or 'infectious' (denoted by 'I'), and thus it is also called the SI model. Staniford et al. [19] presented a propagation model for the Code-RedI v2 worm, which is essentially the above classical simple epidemic model.

2) The classical general epidemic model (Kermack-McKendrick model) [15-18] improves the classical simple epidemic model by considering removal of infectious hosts due to patching (installing software designed to fix security vulnerabilities).

3) The two-factor worm model [14] extends the classical general epidemic model by accounting for removal of susceptible hosts due to patching and considering the pair-wise rate of infection as a variable rather than a constant.

4) The discrete-time Analytical Active Worm Propagation (AAWP) model [20] takes into account the time an infectious host takes to infect other hosts, which is an important factor for the spread of worms [21].

Among the above models, all others are continuous-time except the last one, which is discrete-time.

Stochastic propagation models of computer/Internet worms are based on the theory of stochastic processes. All of them are discrete-time in nature. Two typical examples of stochastic propagation models of computer/Internet worms are as follows.

1) Rohloff and Basar [22] presented a stochastic density-dependent Markov jump process propagation model for computer/Internet worms employing the random scanning approach drawn from the field of epidemiology [16][23].

2) Sellke et al. [24] presented a stochastic Galton-Watson Markov branching process model to characterize the propagation of computer/Internet worms employing the random scanning approach.

However, all of the above models are not applicable to computer/Internet worms employing topological scanning. In recent years, based on the survey on modeling the propagation process of computer/Internet worms [25], we have worked on developing new approaches to support understanding topological scanning based worm propagation, in which some progress has been made [26, 27]. The extended logic matrix approach presented in this paper is part of our continuous effort on the project.

In our extended model, we have taken into account removal, due to quarantine, of both susceptible and infectious peers. However, temporal issues, such as the time lag for worms to infect peers and the time spent in quarantining peers, have not been considered intentionally. The paramount objective of these models is to facilitate determining the maximum number of peers in a P2P system that can be infected in an infinite period, which forms another difference between this work and exiting work, in addition to using logic matrix in worm propagation modeling.

III. THE EXTENDED LOGIC MATRIX APPROACH

A. *Logic matrix and operations*

As mentioned previously, using logic matrix in worm propagation modeling forms the major difference between this work and exiting work. Our reasons of using logic matrix include ease of derivation of the propagation model proposed in this paper, and the model's ease of employment.

We extend the definition of matrix to allow variables or constants of logic type as its elements and term such kind of matrix as logic matrix. The values of variables of logic type can only be one of the two constants of logic type: True (denoted by 'T') or False (denoted by 'F'). If a logic matrix has only one row or one column, we can also term it logic row vector or logic column vector, respectively.

We define absolute value of a variable l of logic type (denoted by $|l|$) as 1 when its value is 'T', and 0 when 'F'; and define absolute value of a logic matrix L (denoted by $|L|$) as the total number of its elements with value 'T'. According to the above definitions, the absolute value of a logic matrix L can be worked out by summing the absolute value of its each element l, i.e.,

$$|L| = \sum |l| \ . \tag{1}$$

A logic matrix L can be inverted. The resultant is a logic matrix of the same dimension with its element l_{inv} being the result of logic NOT operation of the corresponding element l of the logic matrix to be inverted. It can be defined mathematically as follows:

$$l_{inv} = \bar{l} \ , \tag{2}$$

where the bar over l indicates logic NOT operation.

Two logic matrices A and B can be added together if and only if their dimensions are the same, i.e., they have the same number of rows and the same number of columns. The resultant $S = A + B$ is a logic matrix of the same dimension with its element s_{ij} (in the i-th row and the j-th column) being the result of logic OR operation of the corresponding elements a_{ij} and b_{ij} of the two logic matrices to be added together. It can be defined mathematically as follows:

$$s_{ij} = a_{ij} + b_{ij} \ , \qquad (3)$$

where the + sign between a_{ij} and b_{ij} indicates logic OR operation.

Two logic row vectors (or logic column vectors) A and B can be multiplied element-by-element if and only if their dimensions are the same, i.e., they have the same number of columns (or number of rows). The resultant $P = AB$ is a logic row vector (or logic column vector) of the same dimension with its element p_{ij} (in the i-th row and the j-th column) being the result of logic AND operation of the corresponding elements a_{ij} and b_{ij} of the two logic row vectors (or logic column vectors) to be multiplied. It can be defined mathematically as follows:

$$p_{ij} = a_{ij} b_{ij} \ , \qquad (4)$$

where $a_{ij}b_{ij}$ indicates logic AND operation of a_{ij} and b_{ij}.

A logic matrix A can be multiplied by another logic matrix B in the manner of traditional matrix multiplication if and only if their inner dimensions are the same, i.e., number of columns of the multiplicand logic matrix (the left one) is equal to number of rows of the multiplier logic matrix (the right one). The resultant $P = AB$ is a logic matrix with the same number of rows as A and the same number of columns as B. We define value of element p_{ij} (in the i-th row and the j-th column) of the product as determined by the following equation:

$$p_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \ , \qquad (5)$$

where $a_{ik}b_{kj}$ indicates logic AND operation of a_{ik} and b_{kj}, n denotes inner dimensions of the multiplicand and the multiplier logic matrices, and $\sum$ denotes logic OR operation of all resultants of those logic AND operations.

Contrary to logic matrix addition and logic vector multiplication, mutation law does not apply to logic matrix multiplication in the manner of traditional matrix multiplication.

B. Topology logic matrix, state logic vector, and quarantine logic vector of a P2P overlay network

According to the traditional directed graph theory, a P2P overlay network can be represented by a directed graph G, with its set of vertices V representing all peers connected to form the network, and its set of directed edges E representing all directed links among these peers. A directed link from peer i to peer j means peer j is a neighbor of peer i, but peer i is not a neighbor of peer j if there does not exist a directed link from peer j to peer i at the same time. A peer is only able to send messages to its neighbors directly.

Topology of a P2P overlay network consisting of n peers can be represented by an n by n square matrix T with its element t_{ij} (in the i-th row and the j-th column) indicating whether there is a directed link from peer i to peer j. In this paper, we propose a different approach from that used under the traditional directed graph theory to indicating the existence or not of a directed link. The logic constant ‘T’ is used to indicate there is a directed link, and the logic constant ‘F’ to indicate there is not. Therefore, topology of a P2P overlay network consisting of n peers can be represented by an n by n logic square matrix. We term it topology logic matrix of the P2P overlay network.

Each row of the topology logic matrix of a P2P overlay network forms a logic row vector, which is a logic vector representation of outbound links (neighbors) of a particular peer belonging to the network. We call this logic row vector the peer’s topology out-degree logic vector. Each column of the topology logic matrix of a P2P overlay network forms a logic column vector, which is a logic vector representation of inbound links of a particular peer belonging to the network. We call this logic column vector the peer’s topology in-degree logic vector. For example, the i-th row of a topology logic matrix represents all outbound links (neighbors) of peer i; and the j-th column of the topology logic matrix represents all inbound links of peer j.

It can be easily derived that values of topology in-degree and topology out-degree of each peer belonging to a P2P overlay network equate to the absolute values of the peer’s topology in-degree logic vector and topology out-degree logic vector, respectively, which can be worked out by using (1).

Next, we represent states of all the n peers belonging to the P2P overlay network by a logic row vector S of length n with its element s_j (the j-th element) indicating whether peer j has been infected by the worm and become infectious. The logic constant ‘T’ is used to indicate a peer has been infected and become infectious, and the logic constant ‘F’ to indicate it has not. We term the above logic row vector the P2P overlay network’s state logic vector.

It can be easily derived that the total number of infected and infectious peers in a P2P overlay network equates to the absolute value of the network’s state logic vector, which can be worked out by using (1).

Finally, we represent quarantine status of all the n peers belonging to the P2P overlay network by a logic row vector Q of length n with its element q_j (the j-th element) indicating whether peer j has been quarantined for the worm. A quarantined healthy peer will not be infected by the worm; and a quarantined infected and infectious peer will be cured and will not be infected again by the worm. The logic constant ‘T’ is used to indicate a peer has been quarantined, and the logic constant ‘F’ to indicate it has not. We term the above logic row vector the P2P overlay network’s quarantine logic vector.

It can be easily derived that the total number of quarantined peers in a P2P overlay network equates to the absolute value of the network's quarantine logic vector, which can be worked out by using (1).

C. The extended logic matrix approach for modeling propagation of P2P worms

The derivation of our proposed novel logic matrix approach to modeling the propagation of P2P worms is based on the following assumptions:

1) An infected and infectious peer will send the worm packets to all other peers belonging to the same P2P overlay network to which it has a outbound link, regardless of the state (infected by the worm and infectious or not) and the quarantine status (quarantined for the worm or not) of those peers. A healthy (not infected by the worm and not infectious) peer will be infected by the worm and become infectious once it receives the worm packets from an infectious peer, provided the healthy peer is not quarantined for the worm. An infected and infectious peer will remain in that state once it receives the worm packets from an infectious peer, provided the infected and infectious peer is not quarantined for the worm. A healthy peer quarantined for the worm will not be infected by the worm; and an infected and infectious peer quarantined for the worm will be cured and will not be infected again by the worm.

2) The time lags from sending the worm packets, to receiving the worm packets, to having the recipient peers infected by the worm, to the peers infected by the worm becoming infectious will not be considered, nor will the time spent in quarantining peers.

3) There are a total of n peers belonging to a logical (not physical) P2P overlay network under consideration. Initially, there are a total of I_0 peers which are infected by the worm and infectious.

According to the above assumptions, the logical P2P overlay network's initial state (State 0) can be represented by its initial state logic vector S_0 of length n; and the absolute value of S_0 equates to the total number of peers which are initially infected by the worm and infectious (I_0), i.e.,

$$\left| S_0 \right| = I_0 \ . \qquad (6)$$

Generally, State g of the logical P2P overlay network can be represented by its state logic vector S_g of length n; and the absolute value of S_g equates to the total number of peers which are infected by the worm and infectious at that state (I_g), i.e.,

$$\left| S_g \right| = I_g \ . \qquad (7)$$

The next state (State g+1) of the logical P2P overlay network can be represented by its state logic vector S_{g+1} of length n; and the absolute value of S_{g+1} equates to the total number of peers which are infected by the worm and infectious at that state (I_{g+1}), i.e.,

$$\left| S_{g+1} \right| = I_{g+1} \ . \qquad (8)$$

We notice that the logical P2P overlay network's next state represented by its state logic vector S_{g+1} is fully determined by the network's current state represented by its state logic vector S_g, the network's topology represented by its topology logic matrix T, and the network's quarantine status represented by its quarantine logic vector Q. We find the relationship among S_{g+1}, S_g, T, and Q can be described mathematically as follows:

$$S_{g+1} = S_g + \left[\left(S_g T \right) \overline{Q} \right] . \qquad (9)$$

Let S_{gnew} stand for the second term in the above equation (after the + sign), the above equation can be simplified to

$$S_{g+1} = S_g + S_g^{new} \ . \qquad (10)$$

The term represented by Sgnew actually says if at State g at least one peer among those peers from which peer j has inbound links is infectious, peer j will be infected by the worm and become infectious at State g+1 provided peer j is not quarantined.

Since both S_g and Q are logic row vectors of length n and T is an n by n logic square matrix, S_{gnew} will be a logic row vector of length n. It can be derived that S_{gnew} is a logic vector representation of all those peers that can be infected by the worm at State g+1, given the network's state at State g represented by its state logic vector S_g, the network's topology represented by its topology logic matrix T, and the network's quarantine status represented by its quarantine logic vector Q. S_{gnew} may or may not include peer or peers infected by the worm at states prior to State g+1. Then, (9) and (10) can be easily derived.

When quarantined is not enforced at all, (9) will be simplified to

$$S_{g+1} = S_g + \left(S_g T \right) . \qquad (11)$$

Equation (11) is a special case of (9) when is a logic row vector with all its elements being 'T' because Q is a logic row vector with all its elements being 'F'.

Equations (9) and (11) are actually discrete-time deterministic propagation models of P2P worms under quarantine and under no quarantine, respectively, written in the form of difference equations of logic matrix.

Starting from some certain state, there will be no newly infected peer to occur and thus actually, the propagation will stop. The state from which the propagation will cease is the earliest state whose state logic vector S_G satisfies the following equation:

$$\left| S_{G+1} \right| = \left| S_G \right| \ , \qquad (12)$$

where S_{G+1} stands for the state logic vector of the state immediately after the state with state logic vector S_G.

The innovative logic matrix approach proposed above essentially translates the propagation processes of P2P worms into a sequence of logic matrix operations, which can be implemented easily with any matrix-friendly mathematics programs such as MathWorks' MATLAB. It is this feature of the approach that facilitates its applications in the research of the propagation characteristics of P2P worms.

IV. APPLICATIONS OF THE PROPOSED APPROACH

A. *Evaluation metrics*

Our attack-related evaluation metric in this paper is a P2P worm's final infection or coverage (denoted by *c*) in a logical P2P overlay network. It is defined as the number of peers belonging to the network that can be infected by the worm; and can be worked out by using the following equation:

$$c = \left|S_G\right| , \tag{13}$$

where S_G is the state logic vector of the network when the propagation process has just stopped.

Our defense-related evaluation metric in this paper is quarantine rate or ratio (denoted by *q*) for a P2P worm of a logical P2P overlay network. It is defined as the ratio in percentage form of the number of peers belonging to the network that are quarantined for the worm to the total number of peers belonging to the network; and can be worked out by using the following equation:

$$q = \frac{|Q|}{n} \times 100\% , \tag{14}$$

where *Q* is the quarantine logic vector for the worm of the network and n is total number of peers belonging to the network.

Our paramount objective is to find a quarantine tactic whose enforcement will lead to a lower attack performance (measured by the attach-related evaluation metric: coverage *c*) at a lower cost of defense effort (measured by the defense-related evaluation metric: quarantine rate *q*).

B. *Topology of structured P2P systems*

Existing P2P systems can be classified into two broad categories, namely structured and unstructured, according to distributions of their topology out-degree. Topology out-degree defines the number of logical neighbors maintained by each peer locally.

In a structured P2P network, topology out-degree d of each peer is a constant. It is characterized by the following probability distribution:

$$\begin{cases} P(d = k) = 1 \\ P(d \neq k) = 0 \end{cases} , \tag{15}$$

The set of equations (15) gives the probability that a randomly selected peer has *k* neighbors.

C. *Settings of simulation experiments*

We apply the proposed logic matrix approach in our simulation experiments using MathWorks' MATLAB. Our applications are limited to the research of structured P2P systems and the simulation experiments are based on the following assumptions:

1) Topology out-degree (*d*) of each peer belonging to the structured P2P system under consideration strictly follows the power law distribution (15). *d* varies from 1 to 3 with step size 1. Neighbors of a peer are randomly selected from all other peers except the peer itself.

2) Peers quarantined are selected according to the quarantine tactic enforced, which are detailed in the next subsection.

3) The number of initially infected peers, which are selected randomly from all peers not quarantined, varies from 1 to 100 with step size 1.

4) We conduct our simulation experiments for *n* (total number of peers belonging to the system) = 1,000. 1,000 peers are sufficient for our simulation experiments since more peers will not generate significantly different results.

Based on the above assumptions, we populate the topology logic matrix of the structured P2P system under consideration by letting the probability that a randomly selected peer has *k* neighbors follow (15). How to populate the quarantine logic vector of the system is detailed in the next subsection. Once it is populated, we can populate the initial state logic vector of the system.

Our simulation experiments include scenarios with no quarantine at all. Experimental results from them set the benchmark to compare to. When there is no quarantine, (11) instead of (9) forms the foundation of our implementation of the proposed logic matrix approach. When quarantine is enforced, our implementation is based on (9). Each of our simulation experiment is repeated 100 times, and then average values of coverage are reported as final results.

D. *Experimental results*

Random quarantine means peers quarantined are randomly selected from all peers. We employ random quarantine in our simulation experiments. We populate the quarantine logic vector of the structured P2P system under consideration by letting each peer have the same probability of being quarantined when this quarantine tactic is enforced. Then, we populate the initial state logic vector of the system. We conduct our experiments for the 6 sets of scenarios with quarantine rate *q* varying from 0% to 50% with step size 10%. A quarantine rate of 0% actually means no quarantine at all. We include no quarantine as a special case of random quarantine, which facilitates comparison of experimental results.

The experimental results from no quarantine at all when topology out-degree is 1, 2 and 3 are shown in Fig. 1, Fig. 2 and Fig. 3, respectively.

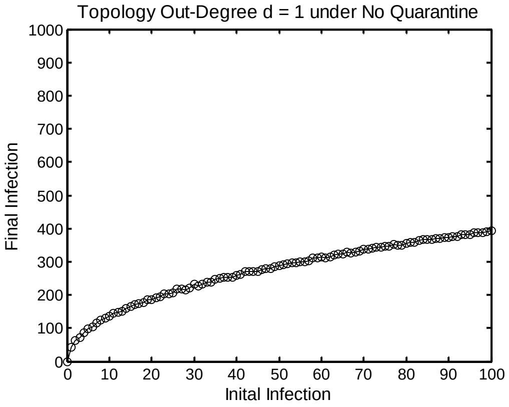


Fig. 1 Final infection as a function of initial infection (quarantine rate = 0%) when topology out-degree is 1.

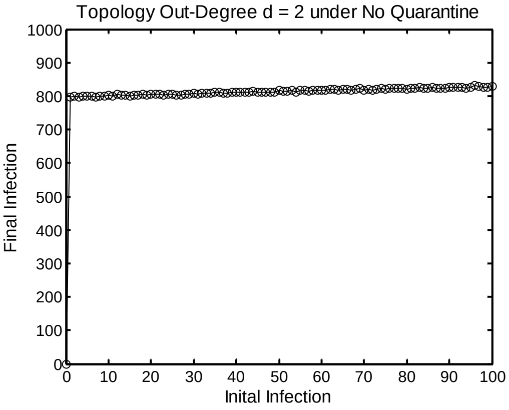


Fig. 2 Final infection as a function of initial infection (quarantine rate = 0%) when topology out-degree is 2.

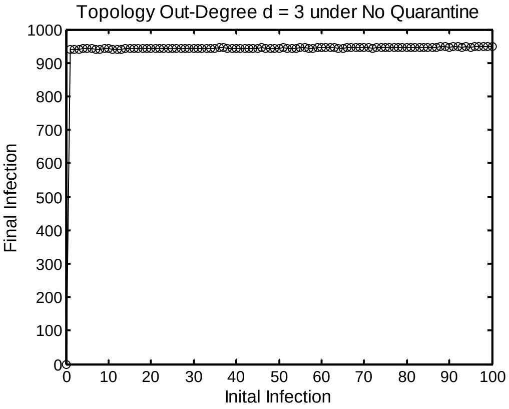


Fig. 3 Final infection as a function of initial infection (quarantine rate = 0%) when topology out-degree is 3.

Fig. 1, Fig. 2 and Fig. 3 reveal that coverage of a P2P worm in a logical P2P overlay network will increase when either topology out-degree or the number of initially infected peers is increased. However, the former has much more significant effect on the coverage than the latter.

The experimental results from random quarantine with varying quarantine rate when topology out-degree is 1, 2 and 3 are shown in Fig. 4, Fig. 5 and Fig. 6, respectively.

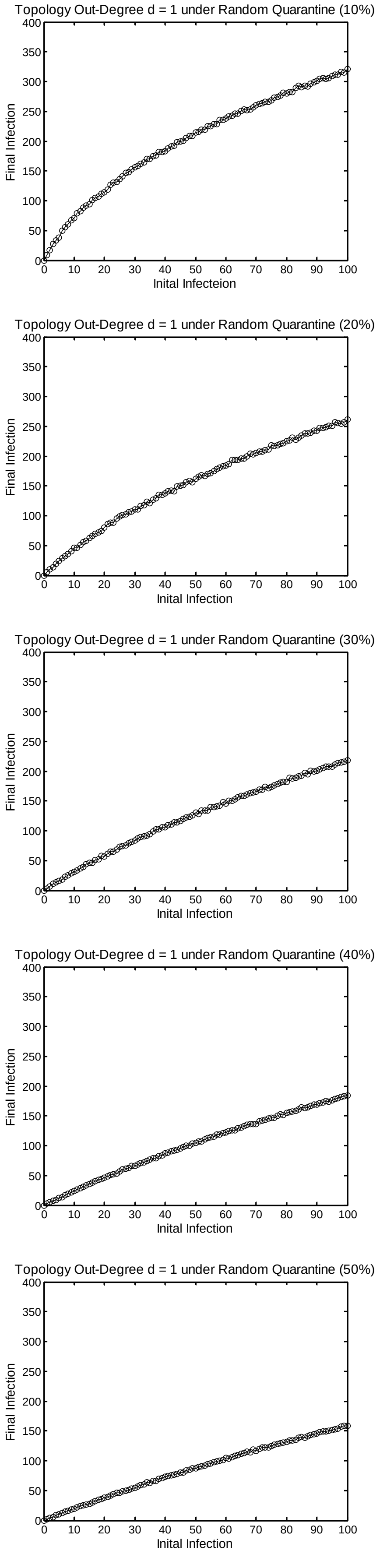


Fig. 4 Final infection as a function of initial infection and quarantine rate when topology out-degree is 1.

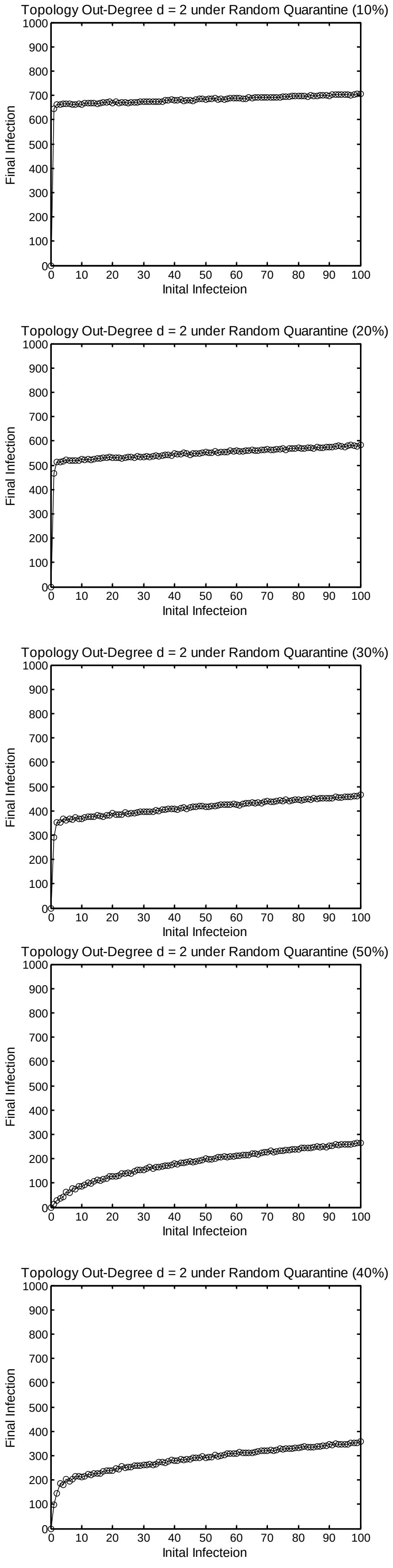


Fig. 5 Final infection as a function of initial infection and quarantine rate when topology out-degree is 2.

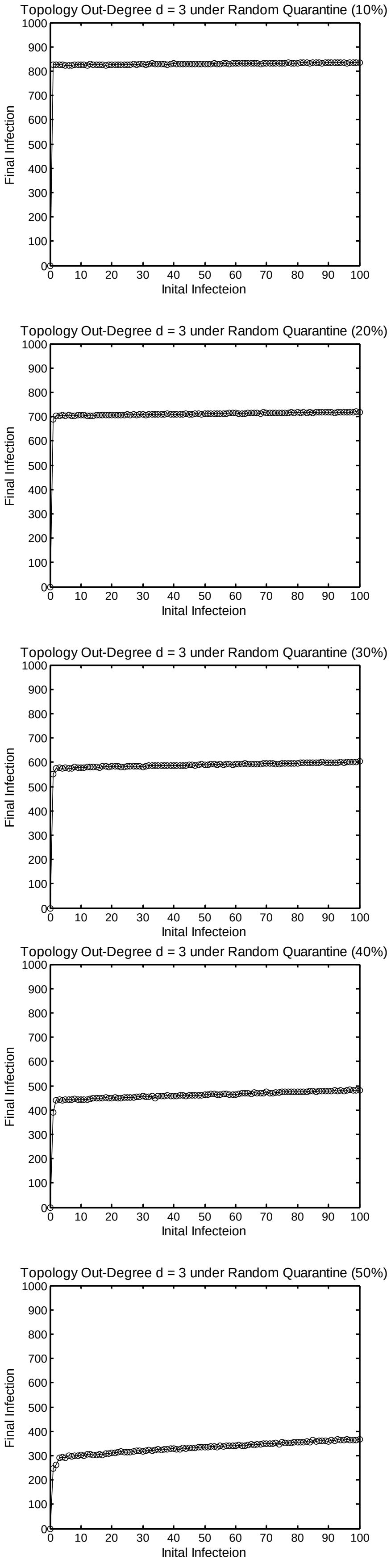


Fig. 6 Final infection as a function of initial infection and quarantine rate when topology out-degree is 3.

Fig. 4, Fig. 5 and Fig. 6 reveal that coverage of a P2P worm in a logical P2P overlay network will decrease when quarantine rate is increased. They also reveal that coverage of a P2P worm in a logical P2P overlay network will increase when either topology out-degree or the number of initially infected peers is increased.

We define quarantine efficiency as a ratio of reduced number of final infection to number of quarantined peers. In other word, quarantine efficiency is defined as the average number of final infection each quarantined peer will reduce. The relationship between quarantine efficiency and quarantine ratio when topology out-degree is 1, 2 and 3 are shown in Fig. 7, Fig. 8 and Fig. 9, respectively.

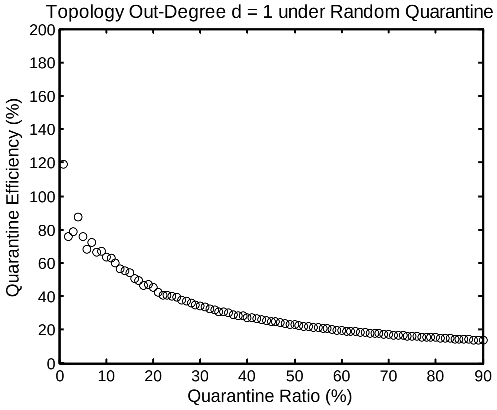


Fig. 7 Quarantine efficiency (%) as a function of quarantine ratio (%) when initial infection ratio is set at 1% and topology out-degree is 1.

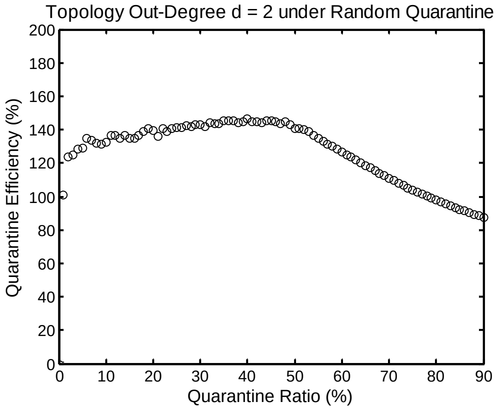


Fig. 8 Quarantine efficiency (%) as a function of quarantine ratio (%) when initial infection ratio is set at 1% and topology out-degree is 2.

Fig. 7, Fig. 8 and Fig. 9 reveal the following. In the case of topology out-degree = 1, maximum quarantine efficiency of approximately 90% is achieved when quarantine ratio is approximately 5%. In the case of topology out-degree = 2, maximum quarantine efficiency of approximately 140% is achieved when quarantine ratio is approximately 40%; however, quarantine efficiency is still greater than 120% when quarantine ratio is approximately 5%. In the case of topology out-degree = 3, maximum quarantine efficiency of approximately 130% is achieved when quarantine ratio is approximately 60%; however, quarantine efficiency is still greater than 100% when quarantine ratio is approximately 5%. Therefore, when we only know topology out-degree is not greater than 3 but do not know its exact value, a quarantine ratio of 5% could be chosen because it will lead to quite, if not most, high quarantine efficiency.

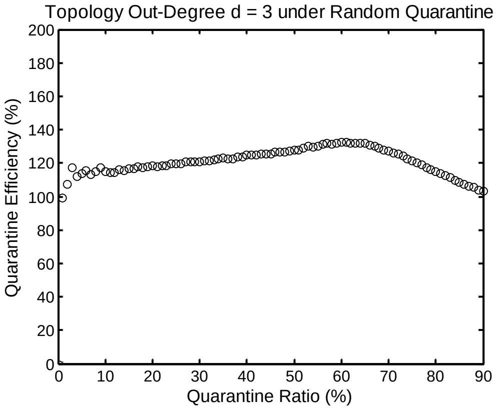


Fig. 9 Quarantine efficiency (%) as a function of quarantine ratio (%) when initial infection ratio is set at 1% and topology out-degree is 3.

V. CONCLUSIONS AND FUTURE RESEARCH

In this paper, based on our extension to matrix and its operations, we propose our logic matrix/vector representations of a P2P overlay network's topology, state, and quarantine status, and present our innovative logic matrix approach to modeling the propagation of P2P worms. We find, from the applications of the approach in our simulation experiments, that coverage of a P2P worm in a structured logical P2P overlay network will increase when either topology out-degree or the number of initially infected peers is increased; and that the former has much more significant effect on the coverage than the latter. It is also found that coverage of a P2P worm in a structured logical P2P overlay network will decrease when quarantine rate is increased. Our most significant finding is that a quarantine ratio of 5% will lead to quite, if not most, high quarantine efficiency when topology out-degree is not greater than 3.

In the future, we are going to look for a more effective and efficient quarantine tactic by applying the proposed discrete- time deterministic propagation model of P2P worms in investigating the impacts of other potential quarantine tactics on the propagation characteristics of P2P worms in a logical P2P overlay network.

REFERENCES

[1] V. Vlachos, S. Androutsellis-Theotokis, D. Spinellis, Security applications of peer-to-peer networks, Computer Networks 45 (2004) 195-205.

[2] V. Pathak, L. Iftode, Byzantine fault tolerant public key authentication in peer-to-peer systems, Computer Networks 50 (2006) 580–597.

[3] W. Yu, S. Chellappan, X. Wang, D. Xuan, Peer-to-peer system-based active worm attacks: modeling, analysis and defense, Computer Communications 31 (2008) 4005–4017.

[4] F. Wang, Y. Zhang, J. Ma, Defending passive worms in unstructured P2P networks based on healthy file dissemination, Computer & Security 28 (2009) 628–636.

[5] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, A taxonomy of computer worms, in Proc. WORM '03, Washington D.C., 2003, pp. 11-18.

[6] D. Moore, C. Shannon, and J. Brown, Code-Red: a case study on the spread and victims of an Internet worm, in Proc. IMW '02, Marseille, France, 2002, pp. 273-284.

[7] C. C. Zou, D. Towsley, and W. Gong, On the performance of Internet worm scanning strategies, University of Massachusetts Technical Report: TR-03-CSE-07, 2003.

[8] C. C. Zou, D. Towsley, W. Gong, and S. Cai, Routing worm: a fast, selective attack worm based on IP address information, in Proc. PADS '05, 2005, pp. 199-206.

[9] Z. Chen and C. Ji, Importance-scanning worm using vulnerable-host distribution, in Proc. IEEE GLOBECOM, 2005, pp. 1779-1784.

[10] Z. Chen and C. Ji, A self-learning worm using importance scanning, in Proc. WORM '05, Fairfax, VA, 2005, pp. 22-29.

[11] E. H. Spafford, The Internet worm program: an analysis, ACM SIGCOMM Computer Communication Review 19 (1989) 17-57.

[12] I. Arce and E. Levy, An analysis of the Slapper worm, in IEEE Security & Privacy, 2003, pp. 82-87.

[13] W. Yu, Analyze the worm-based attack in large scale P2P networks, in Proc. The 8th IEEE International Symposium on High Assurance Systems Engineering (HASE 2004), 2004.

[14] C. C. Zou, W. Gong, and D. Towsley, Code Red worm propagation modeling and analysis, in Proc. CCS '02, Washington D.C., 2002, pp. 138-147.

[15] R. M. Anderson and R. M. May, Infectious diseases of humans: dynamics and control. Oxford: Oxford University Press, 1991.

[16] H. Andersson and T. Britton, Stochastic epidemic models and their statistical analysis. New York: Springer-Verlag, 2000.

[17] N. T. Bailey, The mathematical theory of infectious diseases and its applications. New York: Hafner Press, 1975.

[18] J. C. Frauenthal, Mathematical modeling in epidemiology. New York: Springer-Verlag, 1980.

[19] S. Staniford, V. Paxson, and N. Weaver, How to own the Internet in your spare time, in Proc. Security '02, San Francisco, CA, 2002, pp. 149-167.

[20] Z. Chen, L. Gao, and K. Kwiat, Modeling the spread of active worms, in Proc. IEEE INFOCOM, 2003, pp. 1890-1900.

[21] Y. Wang and C. Wang, Modeling the effects of timing parameters on virus propagation, In Proc. WORM '03, Washington D.C., 2003, pp. 61-66.

[22] K. Rohloff and T. Basar, Stochastic behavior of random constant scanning worms, In Proc. 14th ICCCN, San Diego, CA, 2005, pp. 339-344.

[23] D. J. Daley and J. Gani, Epidemic modelling: an introduction. Cambridge: Cambridge University Press, 1999.

[24] S. Sellke, N. B. Shroff, and S. Bagchi, Modeling and automated containment of worms, In Proc. DSN '05, 2005, pp. 528-537.

[25] Y. Xiang, X. Fan, and W. Zhu, Propagation of active worms: a survey, International Journal of Computer Systems Science & Engineering 24 (2009) 157-172.

[26] X. Fan, Y. Xiang, Modeling the propagation of Peer-to-Peer worms, Future Generation Computer Systems 26 (2010) 1433-1443.

[27] X. Fan, W. W. Guo, and M. Looi, "Modeling and simulating the propagation of unstructured peer-to-peer worms," in 2011 Seventh International Conference on Computational Intelligence and Security, Sanya, China: Los Alamitos, California.: IEEE Computer Society, 2011, pp. 573-577.

Xiang Fan is currently a PhD candidate with School of Engineering and Technology, Central Queensland University in Australia. His research interests include network security in general and modeling and simulating the propagation of computer worms in particular.

William W. Guo received his PhD from The University of Western Australia in 1999. After two years as postdoctoral research fellow at Curtin University and Edith Cowan University in Australia, he has been a faculty member at Edith Cowan University and then Central Queensland University since 2002. He is now a professor and Deputy Dean of School of Engineering and Technology at Central Queensland University, Australia. His research interests include computational intelligence and applications, image processing, data mining and modelling, business intelligence and security. He has published more than fifty papers in international journals, edited books and conference proceedings in these areas. He is a member of IEEE, ACM, and ACS and a regular reviewer for many international journals.

Mark Looi received his PhD from Queensland University of Technology in Australia. He was a professor in security and Head of Computer Science Department at Queensland University of Technology and Dean of School of Information and Commutation Technology at Central Queensland University till July 2012. His research interests include network security, security audits and reviews, security policies, and smart cards. He has supervised many PhD students to completion and published many papers in these areas in international journals, edited books and conference proceedings. He is a member of IEEE and a Fellow of ACS.