# Modeling and Defense against Propagation of Worms in Networks

by

Yini Wang
B.Eng. (Capital Normal University of China)
M.Eng. (Beijing University of Chemical Technology of China)

Submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

Deakin University

March, 2012

**DEAKIN UNIVERSITY**

**ACCESS TO THESIS – A**


I am the author of the thesis entitled

**Modeling and Defense against Propagation of Worms in Networks**

submitted for the degree of

**Doctor of Philosophy**

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

*'I certify that I am the student named below and that the information provided in the form is correct'*

Full Name .....**YINI WANG**.....

Signed ..... Signature Redacted by Library .....

Date .....31/03/2012.....

## DEAKIN UNIVERSITY

### CANDIDATE DECLARATION

I certify that the thesis entitled

**Modeling and Defense against Propagation of Worms in Networks**

submitted for the degree of

**Doctor of Philosophy**

is the result of my own work and that where reference is made to the work of others, due acknowledgment is given.

I also certify that any material in this thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

*'I certify that I am the student named below and that the information provided in the form is correct'*

Full Name .....**YINI WANG**.....

Signed ..... Signature Redacted by Library .....

Date .....31/03/2012.....

# Acknowledgements

I would like to express my sincere gratitude and profound thanks to my supervisor Professor Wanlei Zhou and my associate supervisor Dr. Yang Xiang for their supportive supervision, helpful criticism, valuable suggestions and endless patience. Without their inspiring enthusiasm and encouragement, this thesis would not have been completed. They generously provided me with their time, effort, and insightful advice at all times, and guided me into becoming a successful researcher.

I would like to thank many staff members in the School of Information Technology, Deakin University. They are Professor Lynn Batten, Professor Andrez Goscinski, Dr. Robin Doss, Dr. Shui Yu, Dr. Gang Li, Dr. Ming Li, Dr. Shang Gao, and Dr. Jun Zhang etc. I am also grateful to Ms. Georgina Cahill, Mr. Nghia Dang and other staff in the school for their valuable help.

I would also like to thank my friends and colleagues for their wonderful help with my research and life in general. They are Dr. Ke Li, Dr. Ping Li, Dr. Yiqing Tu, Dr. Faye Ferial Khaddage, Dr. Simon James, Mr. Theerasak Thapngam, Mr. Alessio Bonti, Mr. Sheng Wen, Mr. Min Gan, Mr. Wei Zhu, Ms. Ronghua Tian, Ms. Wei Zhou, Mr. Yongli Ren, Mr. Silvio Cesare, Mr. Yu Wang, Mr. Longxiang Gao, and Ms. Tianqing Zhu.

I cannot finish without thanking my lovely parents, my dad Guangyu Wang and my mum Minli Zhang for their continual support. I would also like to specially thank my elder aunts, Minling Zhang and Mindie Zhang for their encouragement and care.

# Publications

During my PhD Candidature, the following research papers were published or accepted in fully refereed International Conference Proceedings and Journals.

- Sheng Wen, Wei Zhou, Yini Wang, Wanlei Zhou, and Yang Xiang, "Locating Defense Positions for Thwarting the Propagation of Topological Worms", ***IEEE Communications Letters***, accepted 28/02/12, in press. (IF=1.059, ERA 2010=A)

- Yini Wang, Sheng Wen, Silvio Cesare, Wanlei Zhou, and Yang Xiang, "The Microcosmic Model of Worm Propagation", ***The Computer Journal***, Oxford, vol. 54, no. 10, pp. 1700-1720, 2011. (IF=1.327, ERA 2010=A*)

- Yini Wang, Sheng Wen, Silvio Cesare, Wanlei Zhou, and Yang Xiang, "Eliminating Errors in Worm Propagation Models", ***IEEE Communications Letters***, vol. 15, no. 9, pp. 1022-1024, 2011. (IF=1.059, ERA 2010=A)

- Wei Zhou, Sheng Wen, Yini Wang, Yang Xiang, and Wanlei Zhou, " An Analytical Model on the Propagation of Modern Email Worms ", in The 11th IEEE International Conference on Trust Security and Privacy in Computing and Communications (TrustCom-2012), 2012. (ERA 2010=A)

- Yini Wang, Sheng Wen, Wanlei Zhou, and Yang Xiang, "Modeling Worms Propagation on Probability", in The 5th International Conference on Network and System Security (NSS 2011), 2011. (ERA 2010=B)

- Yini Wang, Sheng Wen, Wei Zhou, Wanlei Zhou, and Yang Xiang, "The Probability Model of Peer-to-Peer Botnet Propagation", in The 11th International

Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2011), 2011. (ERA 2010=B)

- Ping Li, Wanlei Zhou and Yini Wang. "Getting the Real-Time Precise Round-Trip Time for Stepping Stone Detection", in The 4th International Conference on Network and System Security (NSS 2010). (ERA 2010=B)

# ABSTRACT

Worms and their variants are widely believed to be one of the most serious challenges in network security research. In recent years, propagation mechanisms used by worms have evolved with the proliferation of data transmission, instant messages and other communication technologies. However, automatically scanning vulnerabilities and sending malicious email attachments (human involvement) are still the two main means for spreading worms.

In order to prevent worms from propagating, as well as to mitigate the impact of an outbreak, we need to have a detailed and quantitative understanding of how a worm spreads. However, previous models mainly focus on analyzing the trends of worm propagation and fail to describe the spreading of worms between different individual nodes. This leads to difficulties in providing a set of optimized and economical patch strategies that deal with the problems of when, where and how many nodes we need to patch. In this thesis, we present a microcosmic analysis of the propagation procedure for scanning worms. It is different from traditional models and can accurately reflect the distribution of nodes in the network in terms of the propagation probabilities. Moreover, from the microcosmic model, we can provide defenders with useful information to answer above three questions and generate a set of optimized patch strategies that minimize the number of infected nodes. The results we obtained can benefit the security industry by allowing them to save significant money in the deployment of their security patching schemes.

The propagation of topology-based worms is a complex procedure as it is closely allied to the topology of the network and requires human interference to spread. Few previous researches have accurately modeled the propagation dynamics of topological worms in an analytical way. Either the spreading speed is overestimated due to the implicit homogeneous mixing assumption or the propagation is investigated through simulation rather than in terms of an analytical model. In this thesis, we propose two methods for modeling the propagation mechanism of typical topology-based worms. In the first method, we use a novel probability matrix to examine the propagation deep inside the spreading procedure among nodes and work out an effective scheme against topology-based worms. In the second method, we derive an accurate propagation

model of email worms by investigating the individual steps and state transitions from an analytical point of view. This not only provides an accurate representation on the propagation of worms with different checking time, but also can reflect the repetitious email sending process. Analysis of experiments demonstrates that the two models are accurate and can aid a better and more realistic understanding of the propagation of topology-based worms.

The thesis is organized as follows. Chapter 1 presents an introduction to the characteristics of worms and their propagation mechanisms, and also describes research objectives and the major contributions of this thesis. Chapter 2 provides a detailed survey of related work carried out in the target discovery techniques of worms and the modeling of worm propagation is also presented. Chapters 3 to 6 present our major contributions for modeling the spreading procedure of scanning worms and topology-based worms. Chapter 3 proposes a microcosmic model of worm propagation by concentrating on the propagation probability and time delay described by a complex matrix. In Chapter 4, we evaluate the microcosmic model for scanning worms and provide a set of optimized and economic patch strategies. In Chapter 5, we propose a novel probability matrix to model the propagation mechanism of a typical topology-based worm, and derive a series of effective defense schemes against it. Chapter 6 presents an analytical model of the propagation dynamics of email worms. Finally, Chapter 7 summarizes the contributions of this thesis and discusses future work.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In this chapter, we begin by introducing the background and basic concepts relevant to this thesis. We then describe the research objectives and highlight the major contributions of the research in our study. Finally, we outline the organization of this thesis.

## 1.1 Background

Worms and their variants have been a persistent security threat in the Internet from the late 1980s, especially during the past decade. For example, the Code Red worm [15] in 2001 infected at least 359,000 hosts in 24 hours and had already cost an estimated $2.6 billion in damage to networks previous to the 2001 attack [86]. The Blaster worm [66] of 2003 infected at least 100,000 Microsoft Windows systems and cost each of the 19 research universities an average of US$299,579 to recover from the worm attacks [87]. Conficker worm [88, 92] was the fifth-ranking global malicious threat observed by Symantec in 2009 and infected nearly 6.5 million computers by attacking Microsoft vulnerabilities. These worms not only lead to large parts of the Internet becoming temporarily inaccessible, but also caused a huge amount of financial loss and social disruption around the world. According to the official Internet

threat report of the Symantec Corporation [59], worms made up the second highest percentage of the top 50 potential malicious code infections for 2009, which rose from 29 percent in 2008 to 43 percent in 2009.

### 1.1.1 Definition of a Worm

A computer worm is a program that self-propagates across a network exploiting security or policy flaws in widely-used services [14]. Worms and viruses are often placed together in the same category, however there is a technical distinction. A virus is a piece of computer code that attaches itself to a computer program, such as an executable file. The spreading of viruses is triggered when the infected program is launched by human action. A worm is similar to a virus by design and is considered to be a sub-class of viruses. It differs from a virus in that it exists as a separate entity that contains all the code needed to carry out its purposes and does not attach itself to other files or programs. Therefore, we distinguish between worms and viruses in that the former searches for new targets to transmit themselves, whereas the latter searches for files in a computer system to attach themselves to and which requires some sort of user action to abet their propagation [89].

### 1.1.2 Worm Categorization

A worm compromises a victim by searching through an existing vulnerable host. There are a number of techniques by which a worm can discover new hosts to exploit. According to the target-search process, we can divide worms into two categories: scan-based worms and topology-based worms.

#### *A. Scan-based Worms*

A scan-based worm (scanning worm) propagates by probing the entire IPv4 space or a set of IP addresses and directly compromises vulnerable target hosts without human interference, such as Code Red I v2 (2001), Code Red II (2001), Slammer/Sapphire (2003), Blaster (2003), Witty (2004) [41], Sasser (2004) [42] and Conficker (2009) [88, 92]. A key characteristic of a scan-based worm is that it can propagate without dependence on the topology. This means that an infectious host is able to infect an arbitrary vulnerable computer.

Scan-based worms employ various scanning strategies, such as random scanning and localized scanning, to find victims when they have no knowledge of where vulnerable hosts reside in the Internet. Random scanning selects target IP addresses randomly, whereas worms using the localized scanning strategy scan IP addresses close to their addresses with a higher probability compared to addresses that are further away.

### *B. Topology-based Worms*

A topology-based worm, such as an email worm and a social network worm, relies on the information contained in the victim machine to locate new targets. This intelligent mechanism allows for a far more efficient propagation than scan-based worms that make a large number of wild guesses for every successful infection. Instead, they can infect on almost every attempt and thus, achieve a rapid spreading speed. Secondly, by using social engineering techniques on modern topological worms, most internet users can possibly fail to recognize malicious codes and become infected, therefore resulting in a wide range of propagation.

A key characteristic of a topology-based worm is that it spreads through topological neighbors. For example, email worms, such as Melissa (1999) [70], Love Letter (2000) [71, 90], Sircam (2001)[91], MyDoom (2004) and Here you have (2010), infect the system immediately when a user opens a malicious email attachment and sends out worm email

copies to all email addresses in the email book of the compromised receiver. For social network worms such as Koobface, the infected account will automatically send the malicious file or link to the people in the contact list of this user.

### 1.1.3 The Propagation of Worms

Worms have attracted widespread attention because they have the ability to travel from host to host and from network to network. Before a worm can be widely spread, it must first explore the vulnerabilities in the network by employing various target discovery techniques. Subsequently, it infects computer systems and uses infected computers to spread itself automatically (as with scan-based worms) or through human activation (as with topology-based worms).

During the propagation of worms, hosts in the network have three different states: susceptible, infectious and removed. A susceptible host is a host that is vulnerable to infection; an infectious host means one which has been infected and can infect others; a removed host is immune or dead so cannot be infected by worms again. According to whether infected hosts can become susceptible again after recovery, researchers model the propagation of worms based on three major models: *SI* models (if no infected hosts can recover), *SIS* models (if infected hosts can become susceptible again) and *SIR* models (if infected hosts can recover). Researchers have also and then presented various defense mechanisms against the propagation of worms.

Although a great deal of research has been done to prevent worms from spreading, worm attacks still pose a serious security threat to networks for the following reasons. Firstly, worms can propagate through the network very quickly by various means, such as file downloading, email, exploiting security holes in software, etc. Some worms can potentially

establish themselves on all vulnerable machines in only a few seconds [7]. Secondly, the rapid advances of computer and network technologies allow modern computer worms to propagate at a speed much faster than human-mediated responses. Thirdly, in order to propagate successfully, worms are becoming more complicated and increasingly efficient. It is therefore of great importance to characterize worm attack behaviors, analyze propagation procedures and efficiently provide patch strategies for protecting networks from worm attacks.

## 1.2 Research Objectives

The objective of this thesis is to model and defend against worm attacks that employ different target discovery techniques. Specifically, we investigate the propagation procedure of worms and aim to provide a set of optimized and economical patch strategies that deal with the following important problems: 1) Where do we patch? 2) How many nodes do we need to patch? 3) When do we patch? In order to address these three questions, we need to model the characteristics of worm propagation that can examine the spreading deep inside the propagation procedure among hosts in the network, ensuring we can accurately understand the spreading and work out effective schemes that minimize the number of infected nodes against the propagation of worms. The results of this research can benefit the security industry by allowing them to save significant money in the deployment of their security patching schemes.

Our research also includes modeling of the propagation dynamics of email worms as they constitute one of the major Internet security problems. We aim to present an analytical model to investigate the details of the propagation mechanisms and characterize the spreading of real-world email worms based on their infection strategies. This model should reflect a

realistic understanding of email worm spreading and provide an accurate representation of the propagation procedure. This analytical model can benefit the creation of new tactics against email worms.

## 1.3 Contributions of the Thesis

In this thesis, we firstly present a microcosmic worm propagation model to accurately access the spreading process and investigate errors which are usually concealed in the traditional macroscopic analytical models. We then apply the proposed microcosmic model to observe the propagation of scan-based worms and provide a series of recommendations and advice for patch strategies to counter worm propagation. We also present a novel process modeling the propagation of topology-based worms to examine the spreading deep inside the propagation procedure and address effective schemes to deal with the problems of where and how many nodes we need to patch. We further model the propagation dynamics of email worms analytically, thus helping us to understand real-world worms based on their different infection methods, which, in turn can benefit the deployment of new defense strategies. The main contributions of our research in this thesis are listed as follows.

### 1.3.1 A Microcosmic Model of Worm Propagation

Existing macroscopic models focus on analyzing the trends of worm propagation and identify very little information within the propagation procedure. These lead to difficulties in dealing with the problems of when, where and how many nodes we need to patch. Therefore, we present a propagation model from a microcosmic view, which is used to examine the spreading deep inside the propagation process of worms between each pair of nodes and can answer the proposed three problems by estimating an optimized patch strategy. We introduce

a complex matrix to represent the propagation probabilities and time delay between each pair of nodes. These two factors result in accurate exploration of the propagation procedure and estimation of both infection scale and the effectiveness of defense. The extension from the real field of the matrix to the complex field of the matrix reflects the mutual effect between these two factors and matches the real case well. This microcosmic model can help evaluate the mutual effect of initial infectious states and patch strategies, and analyze the impact that different distributions of vulnerable hosts have on worm propagation. We create a microcosmic landscape on worm propagation which can provide useful information for a defense against worms.

### 1.3.2 Defense Study against Scanning Worms

We apply the proposed microcosmic model to investigate the propagation procedures of scanning worms. We carry out extensive simulation studies of worm propagation and successfully provide useful information for the proposed problems of where, when and how many nodes we need to patch. According to the results, for high risk vulnerabilities, it is critical that networks reduce the number of vulnerable nodes to below a certain threshold, e.g., 80% in this analysis. We believe the results can benefit the security industry by allowing them to save significant money in the deployment of their security patching schemes. Moreover, through the deployment of different scenarios, we can find how propagation source states, vulnerabilities distributions and patch strategies impact the spreading of worms. In addition, we derive a better understanding of dynamic infection procedures in each step of matrix iteration. These procedures include: 1) What is the propagation probability and time delay between each pair of nodes? 2) How does one node infect another node directly? 3) How does one node infect another node through a group of intermediate nodes? We also discuss the overestimation caused by errors in macroscopic models. Through the analysis of

the propagation procedure, we observe that the error is mainly caused by propagation cycles in the propagation path, which are usually ignored by traditional macroscopic models.

### 1.3.3 Defense Study against Topology-based Worms

An accurate and realistic model of topology-based worms can help us devise effective strategies of defense and reduce expenses for controlling the impact of their outbreak each year. We develop a modeling framework that can characterize the spread of topology-based worms. We first construct the propagation mechanism of topology-based worms by concentrating on the propagation probability and model the propagation procedure through *k*-hops. With the help of the model, we then evaluate the mutual effect of initially infectious states and address effective schemes to deal with the problems of where and how many nodes we need to patch. We take advantage of the propagation probability between each pair of nodes to explore the propagation procedure of worms and estimate both infection scale and defense effectiveness. Through model analysis, we derive a better understanding of dynamic infection procedures in each step rather than recapitulative analysis on propagation tendency. Specifically, we aim to understand: 1) the propagation probability between each pair of nodes; and 2) how one node infects another node through a group of intermediate nodes. From the results, the network administrators can make decisions on how to immunize the highly-connected node to prevent topology-based worm propagation.

### 1.3.4 Modeling the Propagation Dynamics of Email Worms

Modeling the propagation dynamics of email worms not only benefits the development of defense strategies to prevent them from spreading but can also help us investigate the propagation of those isomorphic worms such as Koobface. However, it is hard to provide

mathematical analysis instead of simulation modeling for analyzing the spreading of email worms. The difficulty lies in two aspects: how to characterize the propagation dynamics with different mailbox checking time between email users in a large scale network and how to model the repetitious email sending process for reinfection and self-start reinfection worms. Therefore, an analytical model for observing the spreading procedure of email worms is proposed. We examine the individual spreading steps and every state transition on each node in the network so that our analytical model can reflect the propagation dynamics with the different mailbox checking habits of users. We also propose the concept of virtual users to represent the process of sending repetitive emails so that our analytical model can accurately reflect the propagation of reinfection worms. In addition, our model analyzes the spreading of self-start reinfection worms that most modern email worms belong to and models the repetitious email sending process in self-start reinfection worms. Our evaluation results indicate that our modeling is accurate and can aid a better and more realistic understanding of the propagation of worms. This has potential benefits for devising new tactics against email worms.

## 1.4 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 surveys related work. Chapter 3 presents a new microcosmic worm propagation model that examines deep inside the propagation procedure among individual nodes and is able to provide a series of effective patch strategies against worm propagation. We then apply the proposed microcosmic model to observe the propagation of scan-based worms through the design of different experiments in Chapter 4. Chapter 5 presents a novel process modeling the propagation of topology-based worms by concentrating on the propagation probability. We also analyze the formation of

propagation errors and examine the impact of eliminating errors on the propagation procedure of topology-based worms. In Chapter 6, we focus on modeling the propagation dynamics of email worms analytically. This model studies the propagation procedure of three classes of real-world email worms: non-reinfection, reinfection and self-start reinfection worms. Chapter 7 summarizes the main research contributions and innovations and identifies several possible avenues for future work.

# Chapter 2

# Related Works

This chapter provides an overview of the background and related research on worm propagation. Firstly, we investigate different target discovery mechanisms for two types of worms: scan-based worms and topology-based worms. Then, we study four common topologies of networks for worm spreading. Finally, based on the different spreading strategies and topology information, we provide an analysis and comparison of the current mathematical models typically used to describe worms.

## 2.1 Target Discovery Techniques of Worms

Worms employ distinct propagation strategies such as random, localized, selective and topological scanning to spread. In this subsection, we discuss these target discovery techniques and some of their different sub-classes.

### 2.1.1 Scan-based Techniques

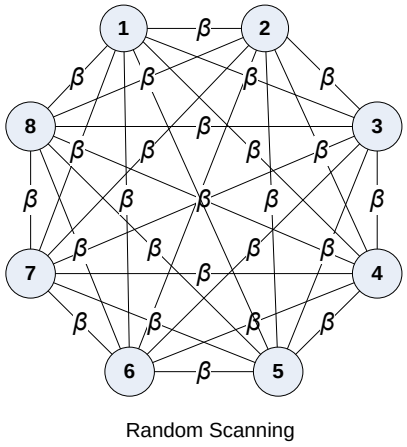



Figure 2.1: Graphical representation of random scanning

Scanning is a very common propagation strategy due to its simplicity and is the most widely employed technique by some well-known scan-based worms such as Code Red, Code Red II, Slammer, Blaster, Sapphire, and Witty worm. Scan-based techniques probe a set of addresses to randomly identify vulnerable hosts or work through an address block using an ordered set of addresses [14].

#### *2.1.1.1 Random Scanning*

Random scanning selects target IP addresses randomly, which leads to a fully-connected topology with identical infection probability $\beta$ for every edge (shown in Fig. 2.1). Several types of scanning strategies, such as uniform, hit-list, and routable scanning, are implemented on the basis of random scanning.

### *A. Uniform Scanning*

Uniform scanning is the simplest strategy to compromise targets when a worm has no knowledge of where vulnerable hosts reside. It picks IP addresses to scan from the whole IPv4 address space with equal probability. This means a worm selects a victim from its scanning space without any preference. Thus, it needs a perfect random number generator to generate target IP addresses at random. Some famous worms, such as Code Red I v1 and v2

[15], and Slammer [12] employed this scanning approach to spread themselves. However, Code-Red I v1 used a static seed in its random number generator and thus generated identical lists of IP addresses on each infected machine. This meant the targets probed by each infected machine were either already infected or impregnable. Consequently, Code-Red I v1 spread slowly and was never able to compromise a high number of hosts. Code-Red I v2 used a random seed in its pseudo-random number generator and thus, each infected computer tried to infect a different list of randomly generated IP addresses. This minor change resulted in more than 359,000 machines being infected with Code-Red I v2 in just fourteen hours [16].

### *B. Hit-list Scanning*

Hit-list scanning was introduced by Staniford *et al*. [7], which can effectively reduce the infection time at the early stage of worm propagation. A hit-list scanning worm first scans and infects all vulnerable hosts on the hit-list, then continues to spread through random scanning. The vulnerable hosts in the hit-list can be infected in a very short period because no scans are wasted on other potential victims. Hit-list scanning hence effectively accelerates the propagation of worms at the early stage. If the hit-list contains IP address of all vulnerable hosts, (called a complete hit-list), it can be used to speed the propagation of worms from beginning to end with the probability of hitting vulnerable or infected hosts equal to 100%. Flash worm [7] is one such worm. It knows the IP addresses of all vulnerable hosts in the Internet and scans from this list. When the worm infects a target, it passes half of its scanning space to the target, and then continues to scan the remaining half of its original scanning space. If no IP address is scanned more than once, then a flash worm is the fastest spreading worm in terms of its worm scanning strategy [5]. Due to bandwidth limitation, however, flash worms cannot reach their full propagation speed. Furthermore, in the real world it is very hard to know all vulnerable hosts' IP addresses. Therefore, complete hit-list scanning is difficult for attackers to implement considering the global scale of the Internet.

### *C. Routable Scanning*

The routable scanning approach probes each IP address from within the routable address space in place of the whole IPv4 address space. Therefore, it needs to determine which IP addresses are routable. Zou *et al*. [4] presented a BGP routable worm as BGP routing tables contain all routable IP addresses. Through scanning the BGP routing table, the scanning address space $\Omega$ of BGP routable worms can be effectively reduced without missing any targets. Currently about 28.6% of the IPv4 address has been allocated and is routable. However, worms based on BGP prefixes have a large payload, which leads to a decrease in the propagation speed. Consequently, a Class A routing worm was presented by Zou *et al*. [4], which uses IPv4 Class A address allocation data. The worm only needs to scan 116 out of 256 Class A address space, which contributes 45.3% of the entire IPv4 space. Routable scanning therefore, improves the spreading speed of worms by reducing the overall scanning space.

#### *2.1.1.2 Localized Scanning*

Instead of selecting targets at random, worms prefer to infect IP addresses that are closer by. Localized scanning strategies choose hosts in the local address space for probing. This leads to a fully-connected topology as shown in Fig. 2.2, where nodes within the same group (group 1 or group 2) infect each other with the same infection probability $\beta_1$, while nodes from different groups infect each other with infection probability $\beta_2$.

### *A. Local Preference Scanning*

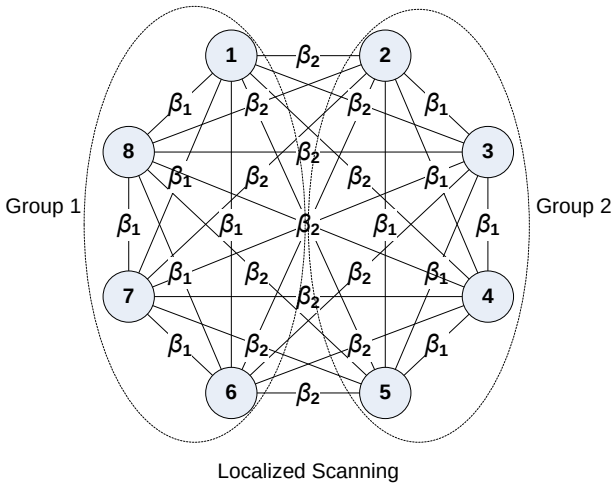



Figure 2.2: Graphical representation of localized scanning

Since vulnerable nodes are not uniformly distributed in the real world, a worm can spread itself quickly when it scans vulnerability dense IP areas more intensively. For this reason, the local preference scanning approach is implemented by attackers, which selects target IP addresses close to a propagation source with a higher probability than addresses farther away. Some localized scanning worms (Code Red II [2,8,9,36] and Blaster worm [10]) propagate themselves with a high probability in certain IP addresses for the purpose of increasing their spreading speed. Taking Code Red II as an example, the probability of the virus propagating to the same Class A IP address is 3/8; to the same Class A and B IP address is 1/2; and to a random IP address is 1/8.

### *B. Local Preference Sequential Scanning*

Different from random scanning, the sequential scanning approach scans IP addresses in order from a starting IP address selected by a worm [5]. Blaster [17] is a typical sequential scan worm because it chooses its starting point locally as the first address of its Class C /24 network with a probability of 0.4 and a random IP address with a probability of 0.6. In selecting the starting point of a sequence, if a close IP address is chosen with higher probability than an address far away, we use the term ‘local preference sequential scanning’.

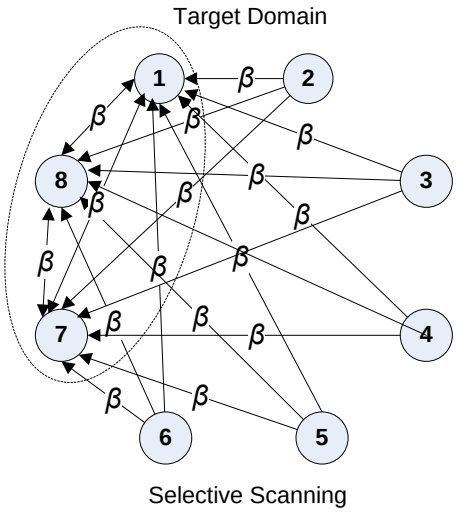



Figure 2.3: Graphical representation of selective scanning

According to an analysis in [5], a worm employing a local preference sequential scanning strategy is more likely to repeat the same propagation sequence, which results in wasting most of the infection power of infected hosts. Consequently, the local preference sequential scanning approach slows down the spreading speed in the propagation of worms.

### *C. Selective Scanning*

Selective scanning is implemented by attackers when they plan to intentionally destroy a certain IP address area rather than the entire Internet, that is, the scanning space is reduced to those selected IP addresses. The selective scanning strategy can lead to an arbitrary topology as shown in Fig. 2.3, where node 4 scans nodes 1, 2 and 3 with infection probability $\beta_1$ and node 5 scans nodes 4, 6, 7 and 8 with infection probability $\beta_2$. If a worm only scans and infects vulnerable hosts in the target domain, it is referred to as *Target-only* scanning. In selective scanning, attackers care more about the spreading speed of a worm in the target domain than the scale of the infected network. According to the analysis in [5], target-only scanning can accelerate the propagation speed if vulnerable hosts are more densely distributed in the target domain.

### 2.1.2 Topology-based Techniques

Topology-based (or topological scanning) techniques are mainly used by worms spreading through topological neighbors. This strategy can lead to an arbitrary topology, as shown in Fig. 2.4, where node $N_i$ ($i$=1,2,…,8) scans its neighbors with a different infection probability $\beta_i$ ($i$=1,2,…,10). A typical example of worms that employ topology-based techniques to launch attacks are email worms. When an email user receives an email message and opens the malicious attachment, the worm program will infect the user's computer and send copies of itself to all email addresses that can be found in the recipient's machine. The addresses in the recipient's machine disclose the neighborhood relationship. Melissa [28] is a typical email worm which appeared in 1999. It looks through all Outlook address books and sends a copy of itself to the first 50 individuals when an infected file is opened for the first time. After Melissa, email worms have become annoyingly common, completed with toolkits and improved by social engineering, such as Love letter in 2000, Mydoom in 2004 and W32.Imsolk in 2010. Recently, topology-based techniques have been used by some isomorphic worms such as Bluetooth worms [21], p2p worms [18-19], and social networks worms [20]. For example, Koobface [27] spreads primarily through social networking sites. It searches the friend list of a user and posts itself as links to videos on their friend's website. When a user is tricked into visiting the website that hosts the video, they are prompted to download a video codec or other necessary update, which is actually a copy of the worm. Users may have difficulty determining if a link was posted by a friend or the worm.

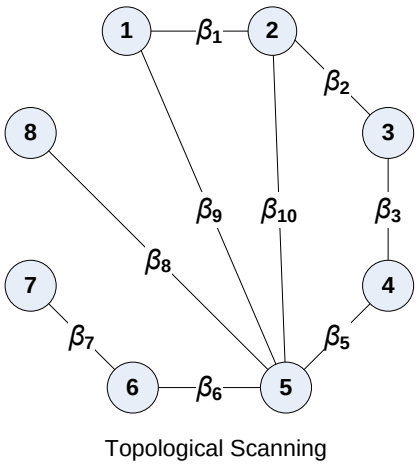



Figure 2.4: Graphical representation of topological scanning

Topology-based techniques utilize the information contained in the victim's machine to locate new targets. This intelligent mechanism allows for a far more efficient propagation than scan-based techniques that make a large number of wild guesses for every successful infection. Instead, they can infect on almost every attempt and thus, achieve a rapid spreading speed. A common feature of topology-based techniques is to involve human interference in the propagation of worms. Taking email worms as an example, the worm program can infect the user's machine and become widespread only when an email user opens the worm email attachment. Thus, whether or not a computer can be infected by malicious emails is determined by human factors including the user's personal habits of checking emails and the user's security consciousness.

## 2.2 Topologies for Modeling the Propagation of Worms

The topology of a network plays a critical role in determining the propagation dynamics of a worm. In the research of epidemic modeling, many types of networks (for example, [6, 10, 33-35, 47, 57]) are adopted to study the effect of epidemic propagation. In this subsection, we

will introduce four typical topologies of networks that are widely used in modeling the propagation of worms.

### 2.2.1 Homogenous Networks

In a homogenous network, each node has roughly the same degree. A fully-connected topology, a standard hypercubic lattice, and an Erdös-Renyi (ER) random network are three typical examples of homogeneous networks [35]. The propagation of worms on homogenous networks satisfies the homogenous assumption that any infected host has an equal opportunity to infect any vulnerable host in the network. In real scenarios, most scan-based worms, such as Code Red I, Code Red II, and Slammer, can propagate without any dependence on the properties of the underlying topology. Thus, homogeneous networks are more suitable for modeling the spreading of scan-based worms.

Recently, many researches [2, 5, 7, 10] studied random scanning worms on homogenous networks using differential equation models. These models assume all hosts in the network can contact each other directly and thus, their topologies are treated as fully-connected graphs. Chen *et al*. [10] proposed an analytical active worm propagation (AAWP) model for randomly scanning worms on the basis of homogenous networks. Yan and Eidenbenz [21] present a detailed analytical model that characterizes the propagation dynamics of Bluetooth worms. It assumes all individual devices are homogenously mixed. Zou *et al.* [2] proposed a two-factor worm model to characterize the propagation of the Code Red worm. This model adopts the homogeneous network, that is, they consider worms that propagate without the topology constraint.

### 2.2.2 Random Networks

A random network is a theoretical construct which contains links that are chosen completely at random with equal probability. Using a random number generator, one assigns links from one node to a second node. Random links typically result in shortcuts to remote nodes, thus shortening the path length to otherwise distant nodes [37]. Recent work [38, 39] provided mechanisms to specify the degree distribution when constructing random graphs and further characterize the size of the large connected component. Fan and Xiang [19] investigated the impact of worm propagation over a simple random graph topology. It assumes each host has the same out-degree. Hosts to which each host has an outbound link are randomly selected from all hosts except the host itself. Of course, the degrees of nodes in a random graph may not be all equal. Zou *et al.* [6] studied the email worm propagation on a random graph. The random graph network was constructed with $n$ vertices and an average degree $E[k] \geq 2$. From the analysis of Zou's model, a random graph cannot reflect a heavy-tailed degree distribution and thus, it is not suitable for modeling topology-based worms.

### 2.2.3 Small-World Networks

A small-world network is a type of mathematical graph where most nodes are not neighbors of one another, but can be reached from every other node by a small number of hops or steps. Small-world networks are highly clustered and have a small characteristic path [51]. Some researchers have observed the dynamic propagation of worms on small-world networks. G. Yan *et al.* [22] considered the BrightKite graph to investigate the impact of malware propagation over online social networks. Compared with the random graph, the BrightKite graph [48] has a similar average shortest path length and a smaller clustering coefficient, and thus, it closely reflects a small-world network structure. Zou *et al.* [6]

modeled email worm propagation on a small-world network that has an average degree $E[k]>4$. It firstly constructs a regular two-dimensional grid network and then connects two randomly-chosen vertices repeatedly until the total number of edges reaches $E[k] \bullet n/4$. From the analysis of Zou's model, a small-world network still cannot provide a heavy-tailed degree distribution and thus, is not suitable for modeling topology-based worms.

### 2.2.4 Power-Law Networks

Power-law networks are networks where the frequency $f_d$ of the out-degree $d$ is proportional to the out-degree to the power of a constant $\alpha$: $f_d \propto d^{\alpha}$ [40]. The constant $\alpha$ is called the power-law exponent. In a power-law network, nodes with the maximum topology degree are rare and most nodes have the minimum topology degree. Recent works have shown that many real-world networks are power-law networks such as social networks [33, 46, 49-50], neural networks [45], and the Web [43-44].

Zou *et al.* [6] and Ebel *et al.* [47] investigated email groups and found that they exhibited characteristics of a power-law distribution. The simulation model proposed by Zou *et al.* [6] studied the dynamic propagation of an email worm over a power-law topology. Although email worms spread slower on a power-law topology than small world topology or random topology, the immunization density is more effective on a power-law topology. Fan and Xiang [19] presented a logic 0-1 matrix model and observed the propagation of worms on a pseudo power law topology. Z. Chen and C. Ji [32] constructed a spatial-temporal model and analyzed the impact of malware propagation on a BA (Barabási-Albert) network [44], which is a type of power-law network. W. Fan *et al.* [20] assumed that the node degree of Facebook users exhibits the power-law distribution and constructed the network using two models: the BA (Barabási-Albert) model and the GLP (Generalized Linear Preference) model.

### 2.2.5 Examples of Real World Topologies

Topology properties affect the spread of topology-based worms, which can either impede or facilitate their propagation and maintenance. Existing works [6, 32, 34] show that structures and characters of the network have strong impact on the spreading speed and scale of worms.

The characters of social networks and the impacting of social structures on the propagation of worms have been intensively investigated in many works [22, 33, 102]. Adamic *et al*. [102] found that the network exhibits small-world behavior through studying an early online social network. Mislove *et al*. [33] presented a large-scale measurement study and analysis of the structure of four popular online social networks: Flickr, Orkut, YouTube and LiveJournal. Their results confirm the power-law, small-world and scale-free properties of online social networks. Yan *et al*. [22] studied the BrightKite network and found that the highly skewed degree distributions and highly clustered structures shown in many social networks are instrumental in spreading the malware quickly at its early stage.

The topology of an email network plays a critical role in determining the propagation dynamics of an email worm [6, 47]. Zou *et al*. [6] examined more than 800,000 email groups in Yahoo! and found that it is heavy-tailed distributed, which exhibits the character of power-law networks. Ebel *et al*. [47] studied the topology of email network that constructed from log files of the email server at Kiel University and found that it exhibits a scale-free link distribution and pronounced small-world behavior.

## 2.3 Worm Propagation Models

In the area of network security, worms have been studied for a long time [1, 93-94]. Early works mainly refer to the academic thought on epidemic propagation and thus, models are constructed according to the state transition of each host including *Susceptible-Infectious* (denoted by '*SI*') models [26], *Susceptible-Infectious-Susceptible* (denoted by '*SIS*') models [53], and *Susceptible-Infectious-Recovered* (denoted by '*SIR*') models [34, 54-55]. In the *SI* framework, all hosts stay in one of only two possible discrete states at any time: susceptible or infectious, which ignores the recovery process. The difference between *SIS* models and *SIR* models depends on whether infected hosts can become susceptible again after recovery. If this is the case, we use the term *SIS* model. Otherwise, if a host cannot become susceptible again once it is cured, we use the *SIR* model, where all hosts stay in one of only three states at any time: susceptible (denoted by '*S*'), infectious (denoted by '*I*'), removed (denoted by '*R*').

Currently, many mathematical models [6-7, 10, 19, 21, 24, 95-101] have been proposed for investigating the propagation of scan-based and topology-based worms on the basis of different state transition models. In this subsection, we mainly focus on these mathematical models and analyze their respective advantages and disadvantages.

### 2.3.1 Homogenous Scan-based Model

The homogenous worm propagation model relies on the homogeneous assumption that each infectious host has an equal probability of spreading the worm to any vulnerable peer in a network. Hence, the homogenous model is based on the concept of a fully connected graph and is an unstructured worm model that ignores the network topology. It can accurately characterize the propagation of worms using scan-based techniques to discover vulnerable

targets, such as Code Red [29-30], Code Red II [2], and Slammer [12]. Scan-based worms scan the entire network and infect targets without regard to topological constraints which means that an infectious host is able to infect an arbitrary vulnerable peer. Up to now, many researchers have modeled the propagation procedure of different types of scan-based worms on the basis of the homogenous assumption. The homogenous model can be further divided into two categories: continuous time and discrete time. A continuous time model is expressed by a set of differential equations, while a discrete time model is expressed by a set of difference equations.

#### *2.3.1.1 Continuous-time Model*

### *A. Classical Simple Epidemic Model*

The Classical Simple Epidemic Model [13, 23-26] is a *SI* model. In this model, the state transition of any host can only be $S \rightarrow I$, and it is assumed a host will remain in the 'infectious' state forever once it has been infected by a worm. Denote by $I(t)$ the number of infectious hosts at time $t$; $N$ the total number of susceptible hosts in the network before a worm spreads out. Thus, the number of susceptible hosts at time $t$ is equal to $[N\text{-}I(t)]$. The classical simple epidemic model for a finite population can be represented by the differential equation below:

$$\frac{dI(t)}{dt} = \beta I(t)\left[N - I(t)\right] \tag{2.1}$$

where, $\beta$ stands for the pair-wise rate of infection in epidemiology studies [13]. It represents a ratio of infection from infectious hosts to susceptible hosts. At the beginning, $t$=0, $I(0)$ hosts are infectious, and in the other $[N\text{-}I(0)]$ all hosts are susceptible.

The Classical Simple Epidemic Model is the most simple and popular differential equation model. It has been used in many papers (for example, [2, 5, 7, 10]) to model random scanning worms, such as Code Red [2] and Slammer [12].

### *B. Uniform Scan Worm Model*

If a worm (i.e. Code Red, Slammer) has no knowledge of the distribution of vulnerable hosts in the network, uniformly scanning all IP addresses is the simplest method to spread itself. Once a host is infected by a worm, it is assumed to remain in the infectious state forever. The uniform scan worm model specifies the abstract parameter $\beta$ in the classical simple epidemic model based on information pertaining to the scanning rate and IP space of the network. Denote by $I(t)$ the number of infectious hosts at time $t$; $N$ the total number of susceptible hosts in the network before a worm spreads out. Thus $[N\text{-}I(t)]$ is the number of susceptible hosts at time $t$. Suppose an average scan rate $\eta$ of a uniform scan worm is the average number of scans an infected host sends out per unit of time. Denote by $\delta$ the length of a small time interval. Thus, an infected host sends out an average of $\eta\delta$ scans during a time interval $\delta$. Suppose the worm uniformly scans the IP space that has $\Omega$ addresses, every scan then has a probability of $1/\Omega$ ($1/\Omega << 1$) to hit any one IP address in this scanning space. Therefore, on average, an infected host has probability $q$ to hit a specific IP address in the scanning space during a small time interval $\delta$.

$$q = 1-(1-1/\Omega)^{\eta\delta} \approx \eta\delta/\Omega, \quad 1/\Omega << 1 \tag{2.2}$$

Here, during the time interval $\delta$, the probability that two scans sent out by an infected host will hit the same vulnerable host is negligible when $\delta$ is sufficiently small. Consequently, the number of infected hosts at time $t+\delta$ will be:

$$I(t+\delta) = I(t) + I(t)\cdot[N - I(t)]\eta\delta/\Omega \tag{2.3}$$

Taking $\delta \to 0$, according to the epidemic model (2.1), the uniform scan worm model can be represented by (2.4):

$$\frac{dI(t)}{dt} = \frac{\eta}{\Omega} I(t)[N - I(t)] \tag{2.4}$$

At time $t$=0, $I(0)$ represents the number of initially infected hosts and $[N-I(0)]$ is the number of all susceptible hosts.

Some variants of random scanning worms (hit-list worms [7], flash worms [5, 7], and routable worms [11]) cannot be directly modeled by (2.4). However, through the extension of the uniform scan worm model, the propagation of these variants of worms can be accurately modeled.

Staniford *et al*. [7] introduced a variant of random scanning worms, called the *hit-list worm*. It first scans and infects all vulnerable hosts on the hit-list, then randomly scans the entire Internet to infect others just like an ordinary uniform scan worm. We can assume the vulnerable hosts on the hit-list to be the initially infected hosts $I(0)$ and ignore the compromising time since they can be infected in a very short time [7]. As a result, a hit-list worm can be modeled by (2.4) along with a large number of initially infected hosts determined by the size of the worm's hit-list.

A flash worm is a variant of the hit-list strategy, introduced by Staniford *et al*. [7]. When a flash worm infects a target, it simply scans half of its scanning space as the other half has been passed to the target including the target host. Since it knows the IP addresses of all vulnerable hosts, that is, the size of scanning space $\Omega = N$, which is much smaller than the entire IPv4 address space ($\Omega = 2^{32}$), and because no IP address is scanned more than once, the flash worm could possibly infect most vulnerable hosts in the Internet in tens of seconds. For this reason, the time delay caused by the infection process of a vulnerable host cannot be ignored in modeling the spreading of flash worms. Denote by $\varepsilon$ the time delay, which is the time interval from the time when a worm scan is sent out to the time when the vulnerable host infected by the scan begins to send out worm scans. We assume a flash worm uniformly scans the address list of all vulnerable hosts. Then, based on the uniform scan model (2.4), the flash worm (uniform scanning) can be modeled by (2.5):

$$\frac{dI(t)}{dt} = \frac{\eta}{N} I(t-\varepsilon)\left[N - I(t)\right], \quad I(t-\varepsilon) = 0, \forall t < \varepsilon \tag{2.5}$$

Another variant of random scanning worms is a routable worm. Zou *et al*. [4] found that currently around 28.6% of IPv4 addresses are routable and thus, they presented a BGP routing worm. It uses BGP routing prefixes to reduce the worm's scanning space $\Omega$. When a BGP routing worm uniformly scans the BGP routable space, it can be modeled by (2.4), where $\Omega$ equals 28.6% of all IP addresses.

Zou *et al*. [5] investigated and compared the propagation performance of random scanning worms and their variants (for example, Code Red, a hit-list worm, a flash worm and a BGP routable worm). Assume the number of vulnerable hosts ($N$) is 360 000, and worms have the same scan rate, i.e., $\eta$ = 358/min. Suppose the size of a worm's hit-list is 10 000, that is, $I(0)$=10 000, while Code Red, the flash worm and the BGP routable worm have 10 initially infected hosts, that is, $I(0)$=10. The scanning space for the BGP routable worm is 28.6% of the entire IP address space, while the Code Red worm and the hit-list worm scan all IP addresses $\Omega = 2^{32}$. For the flash worm, the scanning space $\Omega = N$. From the results of the experiment shown in Fig. 2.5, the flash worm is the fastest spreading worm, which finishes infection within 20 seconds, while Code Red finishes infection after around 500 minutes. At the early stage of propagation, because of a large number size of the hit-list, the hit-list worm can infect more vulnerable hosts than Code Red and the BGP routable worm. Compared with Code Red and the hit-list worm, the BGP routable worm has a smaller scanning space and thus, the infection speed of the routable worm is faster.

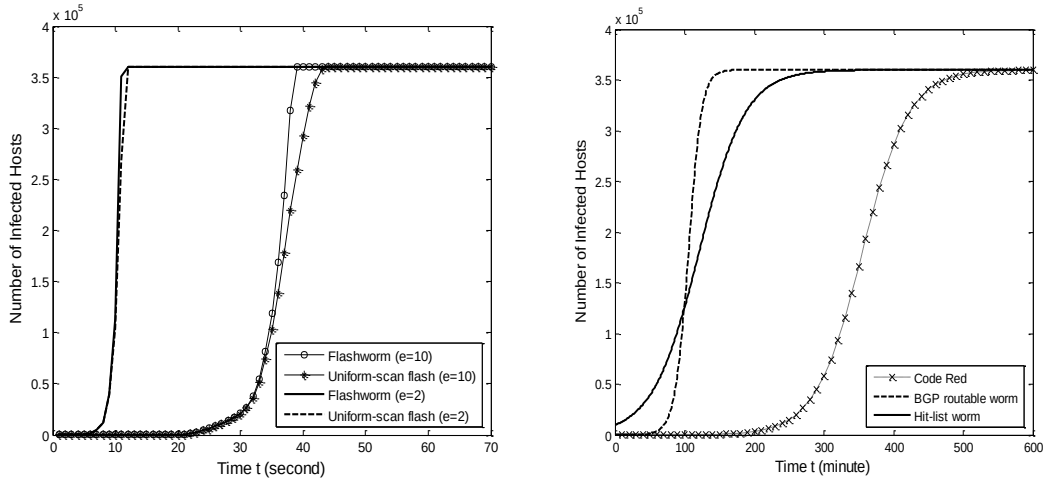



Figure 2.5: Worm propagation of Code Red, BGP routable, hit-list, and flash worm.

### *C. RCS Model*

Staniford *et al*. [7] presented a RCS (Random Constant Spread) model to simulate the propagation of the Code Red I v2 worm, which is almost identical to the classical simple epidemic model. Let $a(t) = I(t)/N$ be the fraction of the population that is infectious at time $t$. Substituting $I(t)$ in equation (2.1) with $a(t)$, and then deriving the differential equation (2.6) below, yields the equation used in [7]:

$$\frac{da(t)}{dt} = ka(t)\left[1 - a(t)\right] \tag{2.6}$$

with solution:

$$a(t) = \frac{e^{k(t-T)}}{1 + e^{k(t-T)}} \tag{2.7}$$

where, $k = \beta N$, and $T$ is a constant of integration that fixes the time position of the incident. Differential equation (2.6) is a logistic equation. For early $t$, $a(t)$ grows exponentially, that is, the number of infectious hosts is nearly exponentially increased at the early stage of worm propagation. For large $t$, $a(t)$ goes to 1 (all susceptible hosts are infected).

### *D. Classical General Epidemic Model: Kermack-McKendrick Model*

Different from the classical simple epidemic model, the Kermack-McKendrick model considered the removal process of infectious hosts [26]. In the Kermack-McKendrick model, all hosts stay in one of only three states at any time: susceptible (denoted by '*S*'), infectious (denoted by '*I*'), removed (denoted by '*R*'). Once a host recovers from the disease, it will be immune to the disease and stay in the 'removed' state forever. The removed hosts can no longer be infected and they do not try to infect others. Therefore, the Kermack-McKendrick model is in the framework of a *SIR* model.

Let $I(t)$ denote the number of infectious hosts at time $t$ and use $R(t)$ to denote the number of removed hosts from previously infectious hosts at time $t$. Denote $\beta$ as the pair-wise rate of infection and $\gamma$ as the rate of removal of infectious hosts. Then, based on the classical simple epidemic model (2.1), the Kermack-McKendrick model can be represented by (2.8):

$$\begin{aligned} \frac{dI(t)}{dt} &= \beta I(t)\left[N - I(t) - R(t)\right] - \frac{dR(t)}{dt} \\ \frac{dR(t)}{dt} &= \gamma I(t) \end{aligned} \tag{2.8}$$

where, $N$ is the size of the finite population. The Kermack-McKendrick model improves the classical simple model by introducing a 'removed' state for each host which means some infectious hosts either recover or die after some time.

### *E. Two-factor Model*

The Kermack-McKendrick model includes the removal of infectious hosts in the propagation of worms, but it ignores the fact that susceptible hosts can also be removed due to patching or filtering countermeasures. Furthermore, in the real world, the pair-wise rate of infection $\beta$ decreases with the time elapsed in the spreading procedure due to the limitation of network bandwidth and Internet infrastructure, while the Kermack-McKendrick model assumes $\beta$ is constant. Therefore, Zou *et al*. [2] introduced a two-factor model, which extends

the Kermack-McKendrick model by considering human countermeasures and network congestion.

In the two-factor model, the removal process consists of two parts: removal of infectious hosts and removal of susceptible hosts. Denote $R(t)$ as the number of removed hosts from the infectious population and $Q(t)$ as the number of removed hosts from the susceptible population. $R(t)$ and $Q(t)$ involve people's security awareness against the propagation of worms. Moreover, in consideration of the slowed down worm scan rate, the pair-wise infection rate $\beta$ is modeled as a function of time $t$, $\beta(t)$, which is determined by the impact of worm traffic on Internet infrastructure and the spreading efficiency of the worm code. Then, the two-factor model can be represented by (2.9):

$$\begin{aligned} \frac{dI(t)}{dt} &= \beta(t)I(t)\left[N - I(t) - R(t) - Q(t)\right] - \frac{dR(t)}{dt} \\ \frac{dR(t)}{dt} &= \gamma I(t) \end{aligned} \tag{2.9}$$

where, $N$ is the finite population size; $I(t)$ denotes the number of infectious hosts at time $t$; $\beta(t)$ is the pair-wise rate of infection at time $t$; and $\gamma$ stands for the rate of removal of infectious hosts. The two-factor model improves the Kermack-McKendrick model through consideration of two major factors that affect worm propagation: human countermeasures like cleaning, patching or filtering and the slowing down of the worm infection rate.

#### *2.3.1.2 Discrete-time Model*

##### *A. AAWP Model*

Chen, Gao and Kwiat [10] presented an AAWP (Analytical Active Worm Propagation) model to take into account the characteristics of random scanning worms spreading according to the homogenous assumption. It assumes that worms can simultaneously scan many machines in a fully-connected network and no hosts can be repeatedly infected. In this model,

active worms scan the whole IPv4 address ($\Omega = 2^{32}$) with equal likelihood, therefore, the probability any computer is hit by one scan is $1/2^{32}$. Denote $m_t$ as the total number of vulnerable hosts (including the infected hosts); denote $n_t$ as the number of infected hosts at time tick $t$ ($t \geq 0$). At time tick $t = 0$, the number of initially vulnerable hosts $m_0$ is equal to $N$ and the number of initially infected hosts $n_0$ is equal to $h$. We suppose $s$ is the scanning rate, and the number of newly infected hosts in each time tick $t$ is equal to $(m_t - n_t)[1-(1-1/2^{32})^{sn_t}]$. Assume that $d$ represents the death rate and $p$ denotes the patching rate. Then, in each time tick the number of vulnerable hosts without being infected and the number of healthy hosts will be $(d + p)n_t$. Therefore, on average in the next time tick $t$+1, the number of total infected hosts can be represented by (2.10):

$$n_{t+1} = n_t + (m_t - n_t)\left[1-\left(1-\frac{1}{2^{32}}\right)^{sn_t}\right] - (d + p)n_t \tag{2.10}$$

In each time tick, the total number of vulnerable hosts including infected hosts is (1-$p$)$m_t$, and thus, at time tick $t$, $m_t$=(1-$p$)$^t m_0$=(1-$p$)$^t N$. Therefore, we can derive (2.11) as follows:

$$n_{t+1} = (1-d-p)n_t + \left[(1-p)^t N - n_t\right]\left[1-\left(1-\frac{1}{2^{32}}\right)^{sn_t}\right] \tag{2.11}$$

where $t \geq 0$ and $n_0$=$h$. Formula (2.11) models the propagation of random scanning worms analytically, and the iteration procedure will stop when all vulnerable hosts are infected or the number of infected hosts remains the same when worms spread.

### *B. Bluetooth Worm Model*

G. Yan and S. Eidenbenz [21] presented a detailed analytical model that characterizes the propagation dynamics of Bluetooth worms. It captures not only the behavior of the Bluetooth protocol but also the impact of mobility patterns on the propagation of Bluetooth worms. This model assumes all individual Bluetooth devices are homogeneously mixed and advances time

in a discrete fashion. Through analyzing a single infection cycle, it derives the duration of an infection cycle $T_{cycle}(t)$ and the number of new infections out of the infection cycle $\alpha(t)$. According to the pair-wise infection rate $\beta(t)$ derived from $\alpha(t)$ and new average density of infected devices at time $t$, this model can estimate the Bluetooth worm propagation curve. From this model, the average density of infected devices in the network at time $t_{k+1}$ is defined by (2.12):

$$i(t_{k+1}) = i(t_k) \cdot \frac{\rho(t_k)}{i'(t_k) + (\rho(t_k) - i'(t_k))e^{-\alpha' \cdot \rho(t_k)/(\rho(t_k) - i'(t_k))}} \tag{2.12}$$

where $i'(t_k)$ is the maximum value between $i(t)$ and $1/S_{inq}(t)$ to ensure at least one infected device in the radio signal covers. $\rho(t_k)$ is the average device density at time $t_k$. Since the worm growth rate can change, and in order to avoid overestimating the number of new infections out of the infection cycle, it uses $\alpha'$ to achieve a better estimation of worm propagation, which is defined by (2.13):

$$\alpha' = \frac{\rho(t_k) - i(t_k)}{\rho(t_k)} \cdot \alpha(t_k) + \frac{i(t_k)}{\rho(t_k)} \cdot \alpha(t_x) \tag{2.13}$$

At the early phase, $\alpha'$ is close to $\alpha(t_k)$ and at the late state of the worm propagation, $\alpha'$ is close to $\alpha(t_x)$. Here, $t_x$ is the latest time when an infected device starts their infection cycle after time $t$ but before time $t_{k+1}$. This model predicts that the Bluetooth worm spreads quickly once the density of the infected devices reach 10 percent, although it propagates very slowly at the early stage.

### 2.3.2 Localized Scan-based Model

Since vulnerable nodes are not uniformly distributed, some localized scanning worms (Code Red II [2, 8-9] and Blaster worm [10]) propagate the virus with a high probability in certain IP addresses for the purpose of increasing their spreading speed. Taking Code Red II

as an example, the probability of the virus propagating to the same Class A IP address is 3/8; to the same Class A and B IP address is 1/2; and to a random IP address is 1/8. Therefore, the localized scanning worm employs a non-homogenous pattern to spread itself in the network. The localized scan-based model can be further divided into two categories: continuous time and discrete time. A continuous time model is expressed by a set of differential equations, while a discrete time model is expressed by a set of difference equations.

#### *2.3.2.1 Continuous-time Model*

### *A. Local Preference Model*

Zou *et al*. [5] took advantage of a continuous time model to describe the spread of localized scanning worms. In this local preference model, it is assumed that a worm has probability $p$ of uniformly scanning IP addresses that have the same first $n$ bits and probability (1-$p$) of uniformly scanning other addresses. Suppose that the worm scanning space contains $K$ networks where all IP addresses have the same first $n$ bits and each network has $N_k$ ($k$=1, 2… $K$) initially vulnerable hosts. Denote by $I_k(t)$ the number of infected hosts in the $k$-th network at time $t$; and denote by $\beta'$ and $\beta''$ the pair-wise rates of infection in local scan and remote scan, respectively. Then we have:

$$\begin{aligned} &\beta' = \frac{p\eta}{2^{32-n}}, \beta'' = \frac{(1-p)\eta}{(K-1)2^{32-n}} \\ &\frac{dI_k(t)}{dt} = \left[\beta' I_k(t) + \sum_{j\neq k}\beta'' I_j(t)\right]\cdot\left[N_k - I_k(t)\right] \end{aligned} \tag{2.14}$$

where $\eta$ represents the average number of scans an infected host sends out per unit of time. Since hosts are not uniformly distributed over the whole Internet, this model supposes only the first $m$ networks ($m$<$K$) have uniformly distributed vulnerable hosts, i.e., $N_1$=…= $N_m$=$N$/$m$, $N_{m+1}$=…=$N_k$=0. Thus, the worm propagation on each network follows (2.15):

$$\frac{dI_k(t)}{dt} = \left[\beta' + (m-1)\beta''\right] \cdot I_k(t)\left[N_k - I_k(t)\right], k = 1,2,...,m \tag{2.15}$$

Suppose $I_k(0) = I_1(0) > 0$, $k$=2,3,…,$m$. We then have:

$$\frac{dI(t)}{dt} = \left[\frac{\beta' + (m-1)\beta''}{m}\right] \cdot I(t)\left[N - I(t)\right] \tag{2.16}$$

(2.14) describes the number of newly infected hosts at time tick $t$ with respect to the entire Internet. This local preference model uses differential equations to reflect the propagation of localized worms that probe different IP addresses with their own preference probabilities.

#### *2.3.2.2 Discrete-time Model*

##### *A. LAAWP Model*

LAAWP (Local Analytical Active Worm Propagation) model is a discrete time model extended from the AAWP model [10]. It characterizes the propagation of worms employing the localized scanning strategy to probe subnets. The worm scans a random address with a probability of $p_0$. For an address with the same first octet, the probability is given by $p_1$, while an address with the same first two octets is scanned with probability $p_2$. In order to simplify the model, both the death rate and patching rate are ignored in the AAWP model. This model assumes localized worms scan a subnet containing $2^{16}$ IP addresses instead of the whole Internet. This subnet is divided into three parts according to the first two octets. Subnet 1 is a special subnet, which has a larger hit-list size. The average number of infected hosts in subnet 1 is denoted $b_1$ and the average number of scans hitting subnet 1 is represented by $k_1$. Subnet 2 contains $2^8$-1 subnets which have the same first octet as subnet 1. The average number of infected hosts in subnet 2 is denoted by $b_2$ and the average number of scans hitting subnet 2 is represented by $k_2$. The other $2^{16}$-$2^8$ subnets belong to subnet 3, which has $b_3$ infected hosts

and $k_3$ scans on average. Therefore, the number of infected hosts in the next time tick is represented by (2.17):

$$b_{i+1} = b_i + \left( \frac{N}{2^{16}} - b_i \right) n_i \left[ 1 - \left( 1 - \frac{1}{2^{16}} \right)^{k_i} \right] \tag{2.17}$$

where $i$ =1, 2, or 3. $k_i$ ($i$=1, 2 or 3) indicates the total number of scans in different subnets coming from the local subnet, the same first octet subnets and the global subnets. The calculation of $k_i$ ($i$=1, 2 or 3) is as follows:

$$k_1 = p_2 s b_1 + p_1 s \left[ b_1 + \left(2^8 - 1\right) b_2 \right] / 2^8 + p_0 s \left[ b_1 + \left(2^8 - 1\right) b_2 + \left(2^{16} - 2^8\right) b_3 \right] / 2^{16}$$
$$k_2 = p_2 s b_2 + p_1 s \left[ b_1 + \left(2^8 - 1\right) b_2 \right] / 2^8 + p_0 s \left[ b_1 + \left(2^8 - 1\right) b_2 + \left(2^{16} - 2^8\right) b_3 \right] / 2^{16}$$
$$k_3 = p_2 s b_3 + p_1 s b_3 + p_0 s \left[ b_1 + \left(2^8 - 1\right) b_2 + \left(2^{16} - 2^8\right) b_3 \right] / 2^{16}$$

The LAAWP model adopts deterministic approximation to reflect the spreading of worms that preferentially scans targets close to their addresses with a higher probability.

### 2.3.3 Topology-based Model

Both homogenous scan-based models and localized scan-based models reflect unstructured worms' propagation without regard to topological constraints. However, a topology-based model describes a structure dependent propagation of worms, which relies on the topology for the spreading of viruses such as email worms [6], p2p worms [18-19], and social network worms [20, 22, 33]. In this subsection, we introduce some typical topology-based discrete-time models.

#### *A. Email Worms Simulation Model*

Zou *et al*. [6] presented a simulation model on the propagation of email worms. It considered the probability of opening an email attachment and email checking frequency, and then compared internet email worm propagation on power law topologies, small world

topologies and random graph topologies. In the proposed model, the probability of each user opening a worm attachment can be treated as an infected probability and the distribution of email checking times can represent the propagation probability.

Due to the high likelihood that email users will also receive email from those they send email to, the Internet's email network is modeled as an undirected graph. According to the distribution of Yahoo! Email groups, authors believe the Internet email network conforms to a heavy-tailed distribution and model the email network topology as a power law network, which follows $F(\alpha) \propto K^{-\alpha}$. The constant $\alpha$ is the power law exponent that determines the degrees of nodes in the network. A larger maximum topology degree requires a larger power law exponent, and a larger expected value of topology degree demands a smaller power law exponent. This model uses $\alpha$ =1.7 to generate the power law network with the total number of hosts $|V|$=100 000 and an average degree of 8. The highest degree for this power law network is 1 833 and the lowest degree is 3.

Email worms depend on email users' interaction to spread. When a user checks an email with a malicious attachment, this user may discard it or open the worm attachment without any security awareness. This user's behavior is represented by an opening probability $C \sim N(0.5, 0.3^2)$ in this model. Then, when a malicious email attachment is opened, the email worm immediately infects the user and sends out a worm email to all email addresses found on this user's computer. Thus, the email checking time is an important parameter that contributes to the propagation speed of the email worm. In this simulation model, the email checking time $T$ follows a Gaussian distribution: $T \sim N(40,20^2)$. This model discusses two cases under different infection assumptions: non-reinfection and reinfection. The main difference is whether a user in the infectious state can be infected again. If the victims can be infected each time they are visited by worms, it is assumed to be a reinfection scenario. Otherwise, infected users send out worm copies only once even if they open a worm attachment again. We refer to this as a

non-reinfection scenario. This email simulation model only considers the propagation of reinfection email worms, which is described as follows.

**Simulation Model:** The discrete-time email worm simulator

```
/* step 1: Initialize parameters */
1. initialize the number of infected nodes infectednum
2. initialize the email checking time CheckingTime and opening probability OpeningProb
   (both follow Gaussian distribution)
3. initialize the number of worm emails: VirusNum, NextVirusNum
4. timetick =1;

/* step 2: Sending worm emails*/
timetick=timetick+1;
for i=1 to the number of total email users do
  if (user i is not HEALTHY or timetick==2)
    if (user i is checking emails)
      if (user i is DANGER)
        user i is INFECTED;
        infectednum=infectednum+1;
      end
      for sendnum=1 to the number of worm emails do
          for link=1 to all the links of user i do
            if (user i opens a worm attachment)
              sending worm emails
            end
          end
      end
      the number of user i's worm email is reset as 0
    end
  end
end

/* step 3: Update Current Node Status */
for i=1 to the number of total email users do
    if (the number of worm emails is not 0)
      if(user i doesn't check the email)
        if(user i is not INFECTED)
          user i is DANGER;
        end
        record the number of worm attachements user i received newly
        reset user i's CheckingTime(i);
      end
    else
        record the total number of worm attachements user i received
    end
    user i's CheckingTime-1;
  end
  Re_InfectedNum(timetick)=infectednum;
```

**end**

According to the discrete-time email worm simulator, the propagation of email worms on a power-law network under the non-reinfection and reinfection scenarios, as shown in Fig. 2.6, illustrate that the spreading speed in the reinfection case is faster and the number of infected hosts at the end of propagation is higher than the non-reinfection case. Based on this simulation model, Zou *et al*. studied the selective immunization defense against email worms. According to their analysis, in a power law topology, if the top 29% of the most-connected nodes are removed from the network, the email network will be broken into separated fragments and no worm outbreak will occur.

### *B. Logic 0-1 Matrix Model*

Fan and Xiang [19] used a logic matrix approach to model the spreading of P2P worms. They presented two different topologies: a simple random graph topology and a pseudo power law topology. The research studied their impacts on a P2P worm's attack performance and analyzed related quarantine strategies for these two topologies.

This model uses a logic matrix (denoted by matrix *T*) to represent the topology of a P2P overlay network. It adopts two constants of logic type (True or 1, False or 0) as the value of matrix variables. The logic constant 'T' indicates the existence of a directed link between two nodes in the network, and the logic constant 'F' is used to indicate there is no directed link. The *i*-th row of a topology logic matrix represents all outbound links of node *i*; and the *j*-th column of the topology logic matrix represents all inbound links of node *j*. This 0-1 matrix stands for the propagation ability of nodes, i.e. whether they can allow the virus to spread or not.

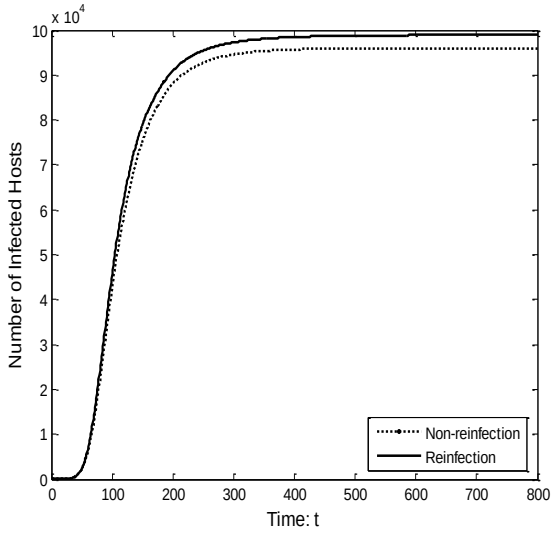



Figure 2.6: Propagation on a power-law network: reinfection vs. non-reinfection.

This logic 0-1 matrix model is a discrete-time deterministic propagation model of P2P worms under three different distributions: infectious state (denoted by logic vector *S*), vulnerability status (denoted by logic vector *V*) and quarantine status (denoted by logic vector *Q*). Where the logic vector $S_g$ represents the current state $g$ of the logical P2P overlay network and the logic vector $S_{g+1}$ represents the next state of the logical P2P overlay network, we have:

$$S_{g+1} = S_g + S_g^{new} \tag{2.18}$$

Here, 1-entries in the vector $S_g^{new}$ represent the transition to infectious at state $g$+1. $S_g^{new}$ varies in consideration of different distributions of *S*, *V*, and *Q*. If all nodes are vulnerable to the worm and no nodes are quarantined, then we have (2.19):

$$S_{g+1} = S_g + S_g T \tag{2.19}$$

If all nodes are not vulnerable to the worm and no nodes are quarantined, then we have (2.20):

$$S_{g+1} = S_g + S_g TV \tag{2.20}$$

If all nodes are vulnerable and some nodes are quarantined, then we have (2.21):

$$S_{g+1} = S_g + S_g T\overline{Q} \tag{2.21}$$

where *Q* bar stands for the distribution of those unquarantined nodes.

This logic 0-1 matrix model translates the propagation processes of P2P worms into a sequence of logic matrix operations. According to the analysis of this model, authors discovered the relation between out-degree, vulnerability and coverage rate in power law topologies and simple random graph topologies respectively, and then proposed quarantine strategies against P2P worms.

### *C. OSN (Online Social Networks) Worms Model*

Fan and Yeung [20] proposed two virus propagation models based on the application network of Facebook, which is the most popular among social network service providers. The difference between email worms and Facebook worms, as the authors highlight, is that people only check if there are any new emails and then log out, while people spend more time on Facebook. In Facebook, two users' accounts appear in each other's friends list if they have confirmed their status to be friends. Thus, the topology of this network is treated as an undirected graph and is constructed by a power-law distribution in the models.

*Facebook application platform based model*: since Facebook provides an application platform that can be utilized by attackers to publish malicious applications, one of the worm propagation models is based on the Facebook application platform. Users of Facebook can install applications to their accounts through this platform. If a user added a malicious application, their account is infected and an invited message is sent to all their friends to persuade them to install the same application, which leads to the spreading of the worm application. The probability of installing one application for user *i* is:

$$P_{user}(i,t) = \frac{AppS_i(t)^{\rho} + init_{user}}{\sum_{j=1}^{N_{user}} \left( AppS_j(t)^{\rho} + init_{user} \right)} \tag{2.22}$$

where $Apps_i(t)$ is the number of applications that user $i$ has installed at time step $t$. The parameter $\rho$ reflects the effect of preferential installation. $init_{user}$ is used to show the initial probability $P_{user}(i,t)$ of a user who does not install any application. Since there are many new installations every day, the probability of one application selected by user $i$ from the application list is:

$$P_{app}(k,t) = \frac{Install_k(t) + init_{user}}{\sum_{j=1}^{N_{app}} \left( Install_j(t) + init_{user} \right)} \tag{2.23}$$

where $init_{app}$ defines the initial probability $P_{app}(k,t)$ of an application without any installation. When a malicious application is installed, invitation messages are sent to all the friends of this infected user. Assuming each user has received $c$ invitations at time step $t$. Then the probability the user is infected is:

$$P_{virus} = \frac{\alpha}{\left( 1 - \frac{Install_{N_{app}}(t)}{N_{user}} \cdot \frac{APPS_i(t)}{N_{app}} \right)^c} \tag{2.24}$$

where $\sigma$ is the percentage of users who accepted the invitations. The infected number $I(t)$ is changed when a malicious application is installed.

*Sending messages based model*: this model investigates the propagation of worms through the sending of messages to friends, which is similar to email worm propagation. When users of Facebook receive malicious emails and click them, these users are infected and worm email copies are sent to their friends. At each time tick, a user can log-in to Facebook with a log-in time $T_{login}(i)$, which follows an exponential distribution. The mean value of $T_{login}(i)$ follows a Gaussian distribution $N(\mu_{Tl}(t),\sigma_{Tl}^2)$. The online time that users spend on Facebook is

$T_{online}(i)$, which follows a Gaussian distribution $N(\mu_{To}(t),\sigma_{To}^2)$. All of the online users may open the malicious email with a probability of $P_{click}$, which follows a Gaussian distribution $N(\mu_p(t),\sigma_p^2)$. The worm propagates until no more new users are infected in the online social network.

### *D. Spatial-temporal Model*

In the work of Chen and Ji [32], a spatial-temporal random process was used to describe the statistical dependence of malware propagation in arbitrary topologies. This spatial-temporal model is a stochastic discrete time model that reflects the temporal dependence and the spatial dependence in the propagation of malware. The temporal dependence means that the status of node *i* (infected or susceptible) at time *t*+1 depends on the status of node *i* at time *t* and the status of its neighbors at time *t*. The temporal dependence of node *i* can be shown as (2.25) and (2.26):

$$P\left(X_i(t+1)=0 \mid X_i(t)=0\right)=\delta_i \tag{2.25}$$

$$P\left(X_i(t+1)=1 \mid X_i(t)=0, \mathrm{X}_{N_i}(t)=\mathrm{x}_{N_i}(t)\right)=\beta_i(t) \tag{2.26}$$

where $X_i(t)$ denotes the status of a network node *i* at time *t* (*t* represents discrete time): if node *i* is infected at time, $X_i(t)=1$; if node *i* is susceptible at time *t*, $X_i(t)=0$. $\mathrm{X}_{Ni}(t)$ is used to denote the status of all neighbors of node *i* at time *t* and the vector $\mathrm{x}_{Ni}(t)$ is the realization of $\mathrm{X}_{Ni}(t)$. If node *i* is susceptible at time *t*, it can be compromised by any of its infected neighbors and become infected at the next time step *t*+1 with a birth rate $\beta_i(t)$. Otherwise, node *i* is infected and has a death rate $\delta_i$ to recover at the next time step *t*+1. The transition probabilities characterize the temporal evolution due to infection and recovery.

Denoting by $R_i(t)$, the probability that node *i* recovers from infected to susceptible status at time *t*+1, is:

$$R_i(t) = P\left(X_i(t+1) = 0, X_i(t) = 1\right) = \delta_i P\left(X_i(t) = 1\right) \tag{2.27}$$

If node *i* is susceptible at time *t*, the probability that node *i* remains susceptible at the next time step can be defined as:

$$\begin{aligned} S_i(t) &= P\left(X_i(t+1) = 0 \mid X_i(t) = 0\right) \\ &= \sum_{\mathrm{x}_{\mathrm{N_i}}(t)} \left[P\left(\mathrm{X}_{N_i}(t) = \mathrm{x}_{\mathrm{N_i}}(t) \mid X_i(t) = 0\right)\left(1 - \beta_i(t)\right)\right] \end{aligned} \tag{2.28}$$

where a joint probability $P(\mathrm{X}_{Ni}(t)=\mathrm{x}_{Ni}(t)|X_i(t)=0)$ representing the status of all neighbors of node *i* at time *t* characterizes the spatial dependence according to the network topology and the interaction between nodes. Based on (2.27) and (2.28), the probability that node *i* is infected at time *t*+1 can be represented by (2.29).

$$P\left(X_i(t+1) = 1\right) = 1 - R_i(t) - S_i(t) P\left(X_i(t) = 0\right) \tag{2.29}$$

Formula (2.24) reflects an iteration process of malware propagation according to the status of a node at time *t* and the status of all neighbors of this node *i* at time *t*, which characterizes the spatial and temporal statistical dependencies. Consequently, the expected number of infected nodes at time *t*, $n(t)$, can be computed:

$$n(t) = E\left[\sum_{i=1}^{M} X_i(t)\right] = \sum_{i=1}^{M} P\left(X_i(t) = 1\right) \tag{2.30}$$

Though (2.24) can be used to study the behavior of malware propagation, the cost of computing $S_i(t)$ is large especially when a node has a great number of neighbors. Therefore, authors presented two models to simplify the challenge posed by the spatial dependence: the Independent Model and the Markov Model.

The *Independent Model* assumes that the status of all nodes at time *t* is spatially independent. This means no propagation cycles are formed when worms propagate via some intermediate nodes because the infected probability of a node is not influenced by its neighbors. Thus, the independent model neglects the spatial dependence. However, the status

of a node at a given time is related to its status at the last time tick and thus, it still remains temporally dependent. The state evolution of node *i* in the independent model can be represented by (2.31):

$$P\left(X_i(t+1)=1\right)=1-R_i(t)-S_i^{ind}(t)P\left(X_i(t)=0\right) \tag{2.31}$$

where

$$S_i^{ind}(t)=\prod_{j\in N_i}\left[1-\beta_{ji}P\left(X_j(t)=1\right)\right]$$

The *Markov Model* assumes that the status of a node is related to its neighbors, but its neighbors cannot be influenced by each other at the same time. This assumption can result in propagation cycles via a single intermediate node, however this can be solved with conditional independence in the network space. If the status of node *i*'s neighbors at the same time step is spatially independent give the status of node *i*, then the state evolution of a node in the Markov model can be represented by (2.32):

$$P\left(X_i(t+1)=1\right)=1-R_i(t)-S_i^{mar}(t)P\left(X_i(t)=0\right) \tag{2.32}$$

where

$$S_i^{mar}(t)=\prod_{j\in N_i}\left[1-\beta_{ji}P\left(X_j(t)=1\mid X_i(t)=0\right)\right]$$

### 2.3.4 Comparison of Worm Propagation Models

A comparison of the various mathematical models of worms discussed above is summarized in Table 2.1. The classical simple epidemic model is the most widely used model for investigating the propagation of scan-based worms using a continuous-time differential equation. Some previous works, such as the uniform scan worm model and the RCS model, are derived from the classical simple epidemic model, which assumes two states for all hosts:

susceptible and infectious, and will stay in the infectious state forever when a host is infected. However, these models are not suitable for cases where the infected and infectious nodes are patched or removed. Consequently, the classical general epidemic model (Kermack-McKendrick model) has been proposed to extend simple epidemic models by introducing a removal process of infectious peers. Continued improvements [2, 56] on modeling worm propagation have considered immunization defense. Zou *et al*. [2] proposed a two-factor worm model, which developed the general epidemic model by taking into account both the effect of human countermeasures and decreases in the infection rate.

The above models adopt a continuous-time differential equation to observe and predict worm spreading in the network. As scanning IP addresses or logical neighbors is usually performed in discrete time [52], a host cannot infect other hosts before it is infected completely. Thus, strictly speaking, the propagation of worms is a discrete event process. A continuous-time model can possibly result in a different spreading speed and infected scale because a host begins devoting itself to infecting other hosts even though only a "small part" of it is infected. Consequently, modeling worm propagation at each discrete time tick is more accurate than using continuous time. The AAWP model, the LAAWP model and the Bluetooth worm model are constructed according to a discrete event process. The AAWP model characterizes the spread of active worms that employ random scanning. LAAWP is extended from the AAWP model and takes into account the characteristics of local subnet scanning worms spreading. The Bluetooth worm model analyzes the propagation dynamics of Bluetooth worms. It captures not only the behavior of the Bluetooth protocol but also the impact of mobility patterns on the propagation of Bluetooth worms.

Table 2.1 A Comparison of Worm Propagation Models

| Worm Propagation Models | Network Topology | Graphical Representation of Topology | Modeling Method | Propagation Process | Model Type | Infection Type |
|---|---|---|---|---|---|---|
| Classical Simple Epidemic Model | H | UG | A | C | SI | Not considered |
| Uniform Scan Worm Model | H | UG | A | C | SI | Not considered |
| RCS Model | H | UG | A | C | SI | Not considered |
| Classical General Epidemic Model | H | UG | A | C | SIR | Not considered |
| Two-factor Model | H | UG | A | C | SIR | Not considered |
| AAWP Model | H | UG | A | D | SIR | Non-reinfection |
| Bluetooth Worm Model | H | UG | A | D | SI | Not considered |
| Local Preference Model | Non-H | UG | A | C | SI | Not considered |
| LAAWP Model | Non-H | UG | A | D | SIR | Non-reinfection |
| Email Worms Simulation Model | R/SW/PL | UG | S | D | SI | Reinfection |
| Logic 0-1 Matrix Model | R/PL | DG | A | D | SIR | Non-reinfection |
| OSN Worms Model | PL | UG | S | D | SI | Non-reinfection |
| Spatial-temporal Model | H/PL | DG | A | D | SIS | Non-reinfection |

H: homogenous mixing; R: random network; SW: small-world network; PL: power-law network;
UG: undirected graph; DG: directed graph;
C: continuous-time event; D: discrete-time event
A: analytical; S: simulation;
SI: susceptible-infected model; SIR: susceptible-infected-recovered model; SIS: susceptible-infected-susceptible model;

All of the above models including continuous-time and discrete-time rely on the homogenous mixing assumption that any infected host has equal opportunity to infect any vulnerable host in the network. However, worms that use a localized scanning strategy, such as Code Red II, require non-homogenous consideration of population locality [7]. Consequently, the local preference model assumes a local preference scanning worm has probability $p$ to uniformly scan addresses which share its first $n$ bits in the network and probability $(1-p)$ to uniformly scan other addresses. Besides, Zou et al. [6] analyzed the propagation of email worms and pointed out that models based on the homogenous mixing assumption overestimate the propagation speed of an epidemic in a topological network, especially in the early stages when a small number of nodes are infected and clustered with

each other. In order to avoid overestimation, the researchers provide a discrete-time simulation model and mainly study the email worm propagation over a power-law topology. This simulation model can more accurately simulate the propagation of email worms than previous homogenous mixing differential equation models. However, this model describes the email worm propagation tendency instead of modeling the dynamic spreading procedure between each pair of nodes. Secondly, they discussed the lower bound for the non-reinfection case, but their model is not capable of accurately eliminating the errors caused by reinfection. Moreover, some assumptions are not realistic. For example, the authors believe that just one malicious email copy will be sent to recipients even if an infected user checks multiple emails containing worms. In reality, a malicious copy is sent whenever the infected user opens a re-infection worm email.

This logic 0-1 matrix model employs a logic matrix to represent links between each pair of hosts and models the spreading of peer-to-peer worms over a pseudo power-law topology. This model can examine the spreading of worms deep inside the propagation procedure among nodes in the network. The model cannot avoid propagation cycles formed among intermediate nodes although it does not allow peers to have outbound links to themselves. These propagation cycles lead to the overestimation in the scale of the infected network. Besides this, their logic matrix is weak regarding an email resembling network because the weight of each link is a probability value ranging from zero to one instead of constant zero or one. The model does not consider the propagation probability and infected probability of each node, which has significant impacts on the infection procedure.

Social networks have become attractive targets for worms. Fan and Yeung [20] proposed the OSN worm model to characterize the behavior of a worm spreading on the application network of Facebook. However, these two models assume a user starts infecting others at every moment once the user is infected. In practice however, infected users spread worms

only as they periodically accept invitations and install malicious applications or check newly received messages and open malicious links. As a result, they have neglected a realistic temporal delay process. Furthermore, the second model simulates the scenario of non-reinfection worm propagation, however non-reinfection worms mainly appear in the early worm cases and are not appropriate for modeling modern email worms that spread over social networks.

The above models assume computer users behave independently, that is, the status of all hosts at the same time step is spatially independent. In real scenarios, however, the propagation of topology-based worms needs human activation and thus the spreading procedure is spatial and temporally dependent. Chen *et al*. [32] used a spatial-temporal random process to describe the statistical dependence of worm propagation in arbitrary topologies. Although this model can outperform the previous models through capturing temporal dependence and detailed topology information, there are also some weak assumptions made. Firstly, this model adopts a *SIS* model, even though infected users are not likely to be infected again after they clean their computers by patching vulnerabilities or updating anti-virus software. Secondly, their model assumes that an infected computer cannot be reinfected. However, recent email worms often reinfect users, and are far more aggressive in spreading throughout the network. Thirdly, the authors ignore an important consideration regarding human behavior; the email checking time, which has been shown to greatly affect the propagation of email worms.

## 2.4 Summary

Worms and their variants are widely believed to be one of the most serious challenges in network security research. Although in recent years propagation mechanisms used by worms

have evolved with the proliferation of data transmission, instant messages and other communication technologies, scan-based techniques and topology-based techniques are still the two main means for the spreading of worms. Modeling the propagation of worms can help us understand how worms spread and enable us to devise effective defense strategies. Therefore, a variety of models have been proposed for modeling the propagation mechanism. This chapter firstly introduced the target discovery techniques for scan-based worms and topology-based worms respectively, illustrating their scanning methods with graphical representations. Secondly, it analyzed the characteristics of four common topologies for modeling worm propagation. Finally, this chapter has described some typical mathematical models of worms that are the analytical tools for investigating dynamics and measuring the propagation of worms. We compared these modes and discussed the pros and cons of each model.

# Chapter 3

# A Microcosmic Worm Propagation Model

Worms and their variants are critical threats to the Internet. Each year, large amounts of money and labor are spent on patching the vulnerabilities in operating systems and various popular software to prevent exploitation by worms. Modeling the propagation process can help us to devise effective strategies against the spread of worms. Most traditional models simulate the overall scale of an infected network for each time tick, making them invalid for examining deep inside the propagation procedure among individual nodes. For this reason, this chapter presents a microcosmic model to analyze worm propagation procedures. Our proposed model can go deep inside the propagation process between each pair of nodes in the network by concentrating on the propagation probability and time delay described by a complex matrix. Moreover, since the analysis gives a microcosmic insight into a worm's propagation, the proposed model can investigate errors which are usually concealed in the traditional macroscopic analytical models. The objectives of this model are to accurately access the spreading and work out an effective scheme against the propagation of worms so the problems of when, where and how many nodes we need to patch can be dealt with.

## 3.1 Introduction

Worms and their variants are widely believed to be one of the most serious challenges in network security research. According to the Symantec Global Internet Security Threat Report [64], the second highest percentage of the top 50 potential malicious code infections for 2009 belonged to worms, which increased from 29 percent in 2008 to 43 percent in 2009. Six of the top 10 threats in 2009 had worm components, compared to only four in 2008. In recent years, propagation mechanisms used by worms have evolved with the proliferation of data transmission, instant messages and other communication technologies.

In order to prevent worms propagating, as well as to mitigate the impact of an outbreak, we need to have a detailed and quantitative understanding of how a worm spreads. Currently, a variety of models have been proposed for modeling the propagation mechanism. Previous work has adopted the classical simple epidemic model [7, 23-25] which simulates two states for all hosts: susceptible and infectious. This is known as the *SI* model. However, this approach is not suitable for cases where the infected and infectious nodes are patched or removed. Consequently, the classical general epidemic model [26, 60], also called the susceptible-infected-recovered (*SIR*) model, has been proposed to extend simple epidemic models by introducing a removal process of infectious peers. Continued improvements [2, 6, 56] on modeling worm propagation have considered the immunization defense. Zou *et al*. [2] proposed a two-factor worm model, which developed the general epidemic model by taking into account both the effect of human countermeasures and decreases in the infection rate. They also studied the propagation model for internet email worms [6] by comparing three different types of topology and summarized the immunization strategies. Although these propagation models perform well in predicting the tendency of worms to spread in the network, macroscopic models identify very little information within the propagation

procedure. This leads to difficulties in dealing with the problems of when, where and how many nodes we need to patch. In fact, there are five parameters involved in modeling worm propagation: *1) propagation probability; 2) infectious nodes' distribution; 3) vulnerable nodes' distribution; 4) patch strategy; 5) time delay.* Previous models of worm propagation have failed to address the following issues:

- *Propagation probability* between each pair of nodes so they cannot locate which set of nodes are more easily infected (Section 3.3.1 and 3.3.2);
- *Propagation time delay* between each pair of nodes so they cannot estimate the time for each node to be infected from the propagation source (Section 3.3.1 and 3.3.2);
- Worms' *propagation procedure* from node to node so they have weak information to decide an appropriate position and time for the patching of each node (Section 3.3.3);
- Errors caused by reinfection in traditional models so they cannot avoid overestimation of patching budget (Section 3.3.4);
- The mutual impact between propagation probability and time delay (Section 3.3.1)

A recent improvement was proposed in [19] which used a logic matrix approach to model the spreading of peer-to-peer worms between each pair of all peers. It adopted two constants of logic type (True or 1, False or 0) as the value of matrix variables. This 0-1 matrix represents the propagation ability of nodes, that is, whether they allow the worms to spread or not. Nevertheless, a significant limitation of this model is that it cannot describe the propagation process of some worms, such as local preference worms, as these worms have different spreading probabilities for specific IP address spaces. More importantly, the model does not include temporal factors, which means it cannot model dynamic worm propagation procedures.

Compared with a macrocosmic propagation model, a microcosmic model can accurately reflect the distribution of nodes in the network, which is beneficial for describing the propagation procedure. We can examine the propagation of worms deep inside the spreading procedure and are able to understand how the current infected states impact on the worm's propagation in the next step. Modeling a microcosmic propagation procedure can provide defenders with useful information to answer the questions of where to patch, how many nodes to patch, and when to patch. Moreover, there is little research in microcosmic propagation models from the view of probability. Therefore, we are motivated to present a microcosmic propagation model for simulating the spreading of worms. Our model has several important components:

- Probability matrixes (PM) are proposed to construct propagation models for worms;
- A Propagation Source vector (*S*) is introduced for describing the distribution of initial infectious nodes;
- A vulnerable distribution vector (*V*);
- A patching strategy vector (*Q*) accounts for a special deployment of patching nodes;
- Propagation abilities (PA).

To the best of our knowledge, there is little research that refers to the microcosmic procedure of worm propagation between nodes in a network. Although research such as [19] analyzed worm propagation from the view of the microcosm, it adopted a simplified logic matrix to indicate the infected states of the network. This simple logic does not effectively describe the propagation procedure between each pair of nodes, nor does it reflect the spreading effect in each step of a worm's propagation.

In order to find an effective and efficient countermeasure against the propagation of worms, we must fully understand their propagation mechanisms. This chapter presents a microcosmic

study on modeling the propagation of worms. The major contributions of this chapter are as follows: firstly, we introduce a complex matrix to represent the propagation probabilities and time delay between each pair of nodes. These two factors lead to accurate exploration of the propagation procedure and estimation of both infection scale and the effectiveness of defense. The extension from the real field of the matrix to the complex field of the matrix reflects the mutual effect between these two factors, which matches the real case well. Secondly, associated with *S*, *V*, *Q*, our model can also help to evaluate: 1) the mutual effect of initial infectious states and patch strategies; 2) the impact different distributions of vulnerable hosts have on worm propagation. Thirdly, we create a microcosmic landscape on worm propagation which can provide useful information for a defense against worms.

We apply our proposed microcosmic model to study the propagation of scanning worms in Chapter 4. Through simulation results, we can derive a set of optimized patch strategies to minimize the number of infected peers and provide economic benefits to industry by selectively deploying security patches.

The rest of this chapter is organized as follows. In Section 3.2, we provide a comparison between macroscopic worm propagation models and microcosmic worm propagation models. In Section 3.3, we model the microcosmic propagation procedure of worms and introduce each component of the proposed model. Finally, we conclude this chapter in Section 3.4 with a brief summary.

## 3.2 Macroscopic and Microcosmic Worm Propagation Models

In the area of network security, both macroscopic [2-10, 15, 29-30, 65-66] and microcosmic [19] models exist for simulating different worm propagation. Most worm propagation models are based on a macroscopic view, such as the homogenous worms'

model, the local preference worms' model and the topological worms' model, which mainly describe the overall spreading tendency of worms. In contrast, microcosm models prefer to study the propagation procedure between nodes according to different scenarios of infectious states, vulnerable states and quarantine states.

## 3.2.1 Macroscopic Worm Propagation Models

### *3.2.1.1 Homogenous Worms' Model*

The homogenous worm propagation model is a simple epidemic model which is used in a lot of research [2, 5, 7-10] to model worm propagation for random scanning worms (Code Red [2], Slammer [12]). Variants of random scanning worms (hit-list worms [7], routable worms [11]) are modeled using extensions of this simple epidemic model. The homogenous model is based on the concept of a fully connected graph and is an unstructured worm model that ignores the network topology. The model assumes each infectious host has an equal probability in spreading the worm to any vulnerable peer in a network. Staniford *et al*. [7] presented an RCS (Random Constant Spread) model to simulate the propagation of the Code-Red I v2 worm, which is almost identical to the homogenous model. Zou *et al*. [2] introduced a two-factor model, which extended the homogenous model by considering human countermeasures and network congestion. These models focus on analyzing the trends of worm propagation. However, they do not describe the worm propagation from node to node or the infection process when disrupted by patching or immunizing nodes. Thus, they are not suitable for modeling the dynamic process of infection and patching between each pair of nodes. Furthermore, the models are significantly limited for modeling worms that scan IP addresses with differing probabilities and are unable to simulate topology-based worm propagation. Additionally, they do not discuss the different impact of reinfection and non-

reinfection on worm propagation. Rohloff and Basar [8] presented a stochastic density-dependent Markov jump process propagation model. Sellke *et al*. [9] provided a stochastic Galton-Watson Markov branching process model. These two models are also limited in simulating the propagation tendency, which is unable to describe the spreading procedure.

#### *3.2.1.2 Local Preference Worms' Model*

Since vulnerable nodes are not uniformly distributed, some localized scanning worms (Code Red II [15, 29-30], Blaster worm [66]) propagate the virus with a high probability in certain IP addresses for the purpose of increasing their spreading speed. Taking Code Red II as an example, the probability of the virus propagating to the same class A IP address is 3/8; to the same class A and B IP address is 1/2; and to the random IP address is 1/8. Thus, the local preference model employs a non-homogenous pattern to simulate worm propagation. Chen *et al*. [10] presented a LAAWP (Local Analytical Active Worm Propagation) model to take into account the characteristics of the spread of local subnet scanning worms. However, this model assumes the distribution of vulnerable hosts is uniform in every subnet. They did not consider the impact of vulnerable distribution on worm propagation, which is one of the important parameters on modeling worms spreading. Zou *et al*. [5] considered the distribution of vulnerable hosts in the IPv4 address space and provided a more accurate method to model the propagation of local preference scanning worms. In this model, they suppose only the first $m$ networks have vulnerable hosts. However, they still assume vulnerability distribution is uniform in each subnet. Moreover, although their model introduced the pair-wise rates of infection in local scanning and remote scanning, it is still derived from the homogenous model. Therefore, these models cannot reflect non-uniform vulnerability distribution on worm propagation and the dynamic process of infection and immunization between each pair of nodes.

#### *3.2.1.3 Topological Worms' Model*

Both the homogenous model and the local preference model reflect the propagation of unstructured worms without regard to topological constraints. However, a topological model describes a structure-dependent propagation of worms, which relies on the topology for the spreading of viruses. Zou *et al*. [6] considered these two probabilities and compared internet email worm propagation on power law topologies, small world topologies and random graph topologies. In the proposed model, the probability of each user opening a worm attachment can be treated as an infected probability and the distribution of email checking times can represent the propagation probability. However, this model still describes the email worm propagation tendency instead of modeling the dynamic spreading procedure between each pair of nodes. In addition, they discussed the lower bound for a non-reinfection case, but their model is not capable of accurately eliminating the errors caused by reinfection.

### 3.2.2 Microcosmic Worm Propagation Models

Microcosmic worm propagation models focus on the infection procedure between each pair of nodes. Fan and Xiang [19] employed a logic matrix approach to model the spreading of peer-to-peer worms between each pair of peers. They discovered the relation between out-degree, vulnerability and coverage rate in power law topologies and simple random graph topologies respectively. However, they did not consider the propagation probability and infected probability of each node, which has significant impacts on the infection procedure. Additionally, although they do not allow peers' outbound links to themselves, they cannot avoid propagation cycles formed among intermediate nodes.

We propose a novel complex matrix that models worm propagation, and simulates the microcosmic spreading procedure of worms. Using this complex matrix in the propagation

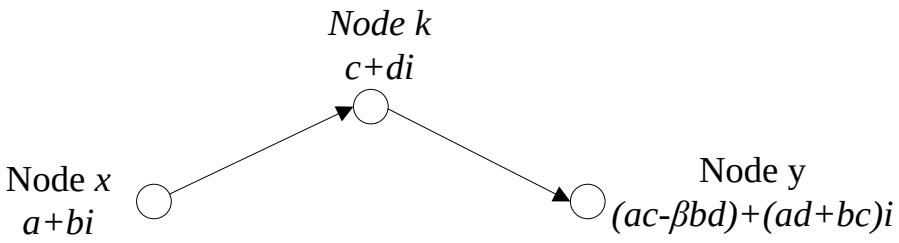



Figure 3.1: Worm propagation computation

simulation of worms forms the major difference between this work and existing work. In our model, we focus on investigating the procedure of worms spreading and providing effective patching strategies, which will benefit IT industries and security best practice.

## 3.3 Propagation Model

In this section we present the propagation model from a microcosmic view, which is used to simulate the propagation process of worms between each pair of nodes and to estimate an optimized patch strategy. We assume that all nodes are vulnerable at the beginning and thus there is no need to scan the whole network.

### 3.3.1 Propagation Matrix (*PM*)

We propose employing an *n* by *n* square complex matrix *PM* with elements $c_{xy}$ to describe a network consisting of *n* peers. We consider that two peers in the network are connected even if the probability of the connection's existence is very small, thereby making node *x* and *y* immediate neighbors. In this matrix, the real component of each element $c_{xy}$ represents the propagation probability of the worm spreading from node *x* to node *y* under the condition that node *x* is infected. The imaginary component represents the propagation delay from node *x* to node *y*. Worms propagating to a target need a certain time delay. If the time delay tends

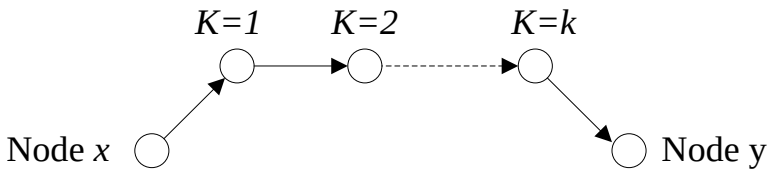



Figure 3.2: Worm propagation between two peers

towards infinity, the target cannot be infected by infectious nodes. Hence, the effect of time delay on a worm's spreading cannot be ignored. The calculation rules of complex can be of benefit to reflect the mutual impact between the propagation probability and time delay (see Section 3.2.2 and Fig. 3.1). However, adopting other means such as two tuples to represent the propagation matrix cannot describe the relation between the above two parameters as each element in the two tuples is separate. We call this complex matrix the propagation matrix (PM) of the network, as shown in (3.1).

$$
\begin{aligned}
&PM = \begin{bmatrix} c_{11} & \cdots & \cdots \\ \cdots & c_{xy} & \cdots \\ \cdots & \cdots & c_{nn} \end{bmatrix}_{n\times n} \\
&c_{xy} = p_{xy} + d_{xy}i, \quad c_{xy} = 0(x = y) \\
&\mathrm{Re}(c_{xy}) = p_{xy} = p(N_y | N_x) \quad p_{xy} \in [0,1] \\
&\mathrm{Im}(c_{xy}) = d_{xy} = t(N_x, N_y) \quad d_{xy} \in (0,1]
\end{aligned}
\tag{3.1}
$$

Each row of the PM represents the propagation probability ($p_{xy}$) and propagation delay ($d_{xy}$) from one infectious peer to all other peers. Each column represents $p_{xy}$ and $d_{xy}$ from infectious peers to a target peer. We assume a peer cannot propagate the worm to itself, so the self-propagation $p_{xx}$ and $d_{xx}$ are zero.

Generally, worms scan an IP address space or a hit-list for propagation. Thus, propagation time delay includes time costs of scanning targets and network latency. Compared with time costs of scanning targets, network latency can be ignored. We assume the imaginary number $i$ as the maximum time cost of scanning the entire IP address space or the hit-list.

### 3.3.2 Propagation Function ($\gamma$)

In real-world conditions, worms could be spread between peers from node $x$ to node $y$ via one or more intermediate nodes, as shown in Fig. 3.2. In existing worms it is observed that an infectious peer can propagate worms and a vulnerable peer can also be infected and become a new infectious node for future propagation. In this scenario, we assume that initially every peer in the PM is vulnerable to the worm.

We assume that worm propagation from node $x$ ($N_x$) to node $y$ ($N_y$) is via and only via $k$ intermediate nodes in a network consisting of $n$ peers. According to the rule of complex multiplication, as shown in Fig. 3.1, the first component ($ac$-$\beta bd$) of the result indicates propagation probability from $N_x$ to $N_y$. Here we manually insert an impact factor ($\beta$) to describe the decrease in the propagation probability caused by time delay. It combines the characteristic of the worm itself and the network it operates on. The second component (($ad$+$bc$)$i$) of the result indicates possible time delay for worm propagation from $N_x$ to $N_y$. It is denoted by $c_{xy}^{(k)}$ and defined in (3.2):

$$\begin{aligned} c_{xy}^{(k)} &= \sum_{m=1}^{m=n,m\neq x} c_{xm}^{(k-1)} c_{my} \\ &= \sum_{m=1}^{m=n,m\neq x} ((p_{xm}^{(k-1)} p_{my} - \beta t_{xm}^{(k-1)} t_{my}) + (p_{xm}^{(k-1)} t_{my} + t_{xm}^{(k-1)} p_{my})i) \\ k &\in [1, n-2], \quad x = 1,...,n, \qquad y = 1,...,n \end{aligned} \tag{3.2}$$

Since $N_x$ self-propagation via $k$ nodes is meaningless in the real world, we define the value of this propagation probability as zero; namely $c_{xy}^{(k)}$ =0 when $x$=$y$. We introduce a function $\gamma$ to conduct the iterated procedure as in (3.3):

$$\begin{aligned} \gamma^0(PM) &= PM \\ \gamma^k(PM) &= \underbrace{PM \times PM \times \cdots \times PM}_{k+1} \end{aligned} \tag{3.3}$$

Operation × is the traditional matrix multiplication. Subsequently, the PM can be represented by the following equation when the worm propagation is via and only via *k* intermediate nodes, as shown in (3.4):

$$PM^{(k)} = \begin{bmatrix} c_{11}^{(k)} & \cdots & \cdots \\ \cdots & c_{xy}^{(k)} & \cdots \\ \cdots & \cdots & c_{nn}^{(k)} \end{bmatrix}_{n \times n} = \gamma^k (PM) \tag{3.4}$$

### 3.3.3 Three Key Factors

In a network, there are three significant factors for worm propagation: infectious state, vulnerability distribution, and patch strategy. The infected state represents the state of whether the peer has been infected or not. Vulnerability distribution identifies vulnerable peers in the network. A patch strategy provides an approach to cure infected peers. Infected peers cannot be infected after being patched.

#### *3.3.3.1 Propagation Source Vector (S)*

An initial propagation source vector (*S*) is defined as shown in (3.5). An infectious peer that can propagate worms is represented with a probability of one. The probability of zero means that a peer is healthy and does not have the ability to propagate the worm.

$$S = [s_1, s_2, \ldots, s_x, \ldots, s_n]^T, \ s_x = 0 \ or \ 1, \quad x = 1 \ldots . n \tag{3.5}$$

The iterated procedure can be represented as function $\gamma_s$ in (3.6):

$$\begin{aligned} \gamma_s^{\,0}(PM) &= S \,\&_L\, PM \\ \gamma_s^{\,k}(PM) &= \gamma_s^{\,k-1}(PM) \times PM \\ &= (S \,\&_L\, PM) \times \underbrace{PM \times \cdots \times PM}_{k} \qquad (k \geq 1) \end{aligned} \tag{3.6}$$

We define $\&_L$ to indicate a new logic AND operation of a column vector *A* and a matrix *B*, called *Left Logic AND*. The result of $A \&_L B$ is a new logic matrix of the same dimension as *B*. This operation is used to eliminate non-infectious nodes. Each element in the new matrix is the result of the product of the corresponding elements $a_x$ and $b_{xy}$ from each column of matrix *B*. It is defined in (3.7):

$$\begin{aligned} A \&_L B &= \begin{bmatrix} a_1 \\ \dots \\ a_n \end{bmatrix} \&_L \begin{bmatrix} b_{11} & \dots & \dots \\ \dots & b_{xy} & \dots \\ \dots & \dots & b_{nn} \end{bmatrix} \\ &= \begin{bmatrix} a_1 \times b_{11} & \dots & a_1 \times b_{1n} \\ \dots & a_x \times b_{xy} & \dots \\ a_n \times b_{n1} & \dots & a_n \times b_{nn} \end{bmatrix} \end{aligned} \tag{3.7}$$

The PM can be represented by the following equation when a worm's propagation is via and only via *k* intermediate nodes, as in (3.8).

$$\begin{aligned} &PM_s^{(0)} = \gamma_s^{\,0}(PM) \\ &PM_s^{(k)} = \gamma_s^{k}(PM) = PM_s^{(k-1)} \times PM \qquad (k \geq 1) \end{aligned} \tag{3.8}$$

During the propagation process, each intermediate node can be infected and become infectious. We introduce an infected state vector *I*, as shown in (3.9):

$$\begin{aligned} &I = [ie_1, ie_2, \dots, ie_x, \dots, ie_n]^T, \ ie_x = 0 \ or \ 1, \quad x = 1 \dots n \\ &I_s^{(k)} = \Gamma(S^T, PM_s^{(k)}), \quad (k \geq 0) \end{aligned} \tag{3.9}$$

where *Γ* function computes each item in infected state vector *I* using the formula as shown in (3.10):

$$ie_x = \sum_y S_y p_{yx}^{(k)} + \frac{\sum_y S_y p_{yx}^{(k)} t_{yx}^{(k)}}{\sum_y S_y p_{yx}^{(k)}} i \tag{3.10}$$

$I_s^{(k)}$ reflects the infected possibility and time delay of each node after worm propagation via *k* intermediate nodes under a certain deployment of *S*.

#### *3.3.3.2 Vulnerable Distribution Vector (***V***)*

Under real-world conditions, the vulnerability of a peer is an objective fact. Therefore, a healthy peer without any vulnerability cannot become infectious in the worm's propagation process. On the basis of this fact, we need to consider the vulnerability distribution in the PM. The vulnerable distribution vector (*V*) is defined in (3.11). For an element in *V*, the value of one represents that a peer is vulnerable. Zero means that the peer is healthy and is not vulnerable.

$$V = [v_1, v_2, \ldots, v_x, \ldots, v_n]^T, \ v_x = 0 \ or \ 1, \quad x = 1 \ldots . n \tag{3.11}$$

Once nodes are vulnerable, they can become infected and have the ability to infect others. Therefore, the iterated procedure can be represented as function $\gamma_{sv}$ in (3.12):

$$\begin{aligned} \gamma_{sv}^{\ 0}(PM) &= S \,\&_L\, PM \,\&_R\, V^T \\ \gamma_{sv}^{\ k}(PM) &= \gamma_{sv}^{\ k-1}(PM) \times (V \,\&_L\, PM \,\&_R\, V^T) \quad (k \geq 1) \end{aligned} \tag{3.12}$$

We define $\&_R$ to indicate a new logic AND operation of a column vector *A* and a matrix *B*, called *Right Logic AND*, which is different from *Left Logic AND*. The result of $A \,\&_R\, B$ is a new logic matrix of the same dimension as *B*. Each element in the new matrix is the result of the product of the corresponding elements $a_y$ and $b_{xy}$ from each row of matrix *B*. It is defined in (3.13):

$$B \&_R A = \begin{bmatrix} b_{11} & \dots & \dots \\ \dots & b_{xy} & \dots \\ \dots & \dots & b_{nn} \end{bmatrix} \&_R \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} = \begin{bmatrix} b_{11} \times a_1 & \dots & b_{1n} \times a_n \\ \dots & b_{xy} \times a_y & \dots \\ b_{n1} \times a_1 & \dots & b_{nn} \times a_n \end{bmatrix} \quad (3.13)$$

Considering the vulnerability distribution vector, the PM and infected probability vector *I* can be represented by the following equations respectively when the worm propagates via and only via *k* intermediate nodes, as in (3.14).

$$\begin{aligned} PM_{sv}^{(k)} &= \gamma_{sv}^{k}(PM) \quad (k \geq 0) \\ I_{sv}^{(k)} &= \Gamma(S^T, PM_{sv}^{(k)}) \quad (k \geq 0) \end{aligned} \quad (3.14)$$

#### *3.3.3.3 Patch Strategy Vector (Q)*

An infected peer can be cured and become a healthy node, unable to spread worms to other peers. Therefore, we need to remove these nodes from the propagation process at that time. We define a patch vector *Q* in (3.15). For each element in *Q*, the value of one represents that a peer has been patched and is now a healthy node. A value of zero indicates that a peer is still vulnerable.

$$Q = [q_1, q_2, \dots, q_x, \dots, q_n]^T, \ q_x = 0 \ or \ 1, \quad x = 1 \dots n \quad (3.15)$$

Once the nodes have been patched, they will become immune to the worms and lose their infectious ability. Thus, we should exclude these patched nodes in the matrix for the successive iteration. The iterated procedure can be represented as function $\gamma_{svq}$ shown in (3.16):

Table 3.1. Truth Table for New Logic And Operation

| $V^T$ | $Q^T$ | $V^T \underline{\&} Q^T$ |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |

$$
\begin{aligned}
&Q' = V \ \underline{\&}\ Q \\
&\gamma_{svq}{}^{0}(PM) = S \,\&_L\, PM \,\&_R\, Q'^{T} \\
&\gamma_{svq}{}^{k}(PM) = \gamma_{svq}{}^{k-1}(PM) \times (Q' \,\&_L\, PM \,\&_R\, Q'^{T}) \quad (k \geq 1)
\end{aligned}
\tag{3.16}
$$

We define $\underline{\&}$ to indicate a new logic AND operation between two elements. The definition for $\underline{\&}$ operation is shown in Table 3.1.

After considering the patch strategy vector, the PM and infected probability vector *I* can be represented by the following equations respectively when the worm propagates via and only via *k* intermediate nodes, as shown in (3.17):

$$
\begin{aligned}
&PM_{svq}{}^{(k)} = \gamma_{svq}{}^{k}(PM) \quad (k \geq 0) \\
&I_{svq}^{(k)} = \Gamma(S^T, PM_{svq}^{(k)}) \qquad (k \geq 0)
\end{aligned}
\tag{3.17}
$$

### 3.3.4 Error Calibration Vector (*E*)

We consider two scenarios of infection: reinfection and non-reinfection. Generally, reinfection means a node can be infected repeatedly and non-reinfection indicates a node can only be infected once [15].

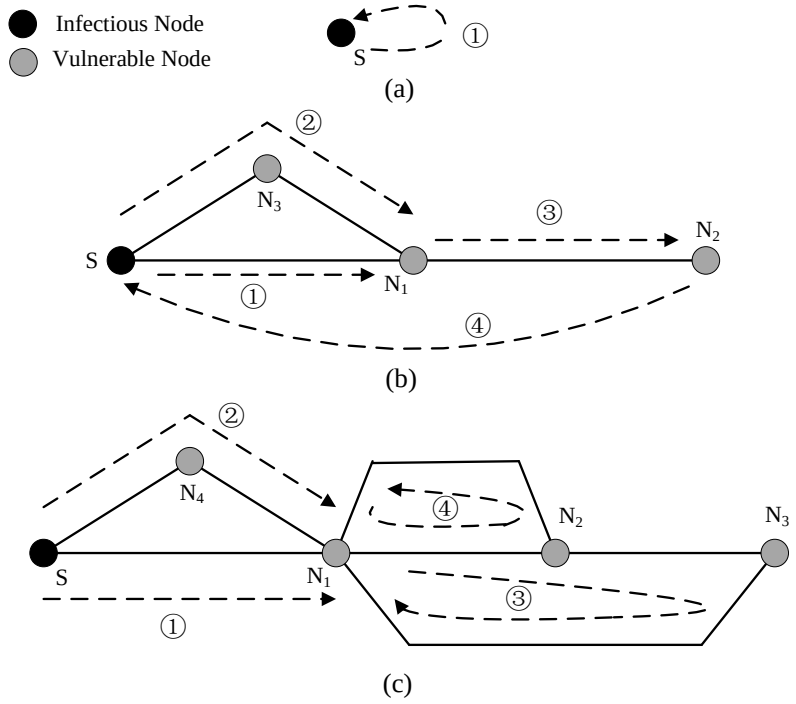



Figure 3.3: Propagation cycles

If a worm belongs to the reinfection type, the earlier-mentioned propagation mechanism is reasonable. However, if a worm belongs to the non-reinfection type, propagation cycles will be formed during the spreading procedure, which results in errors in the infected probabilities, as shown in Fig. 3.3. There are three types of cycles formed in the propagation procedure. As shown in Fig. 3.3(a), an infectious node *s* could spread the worm to itself (*S*-*S*). This is called self-propagation, which results in an increase of infected probability infinitely. In the real case, however, this infectious node can be infected only once. Thus, in our propagation model, we do not allow a self-propagation cycle, that is, each node can be infected by $N$ ($N \geq 1$) nodes and no peer has an outbound link to itself. The work completed by [19] also noticed this self-propagation characteristic and also avoided it.

The second type of cycle is shown in Fig. 3.3(b). The initial infectious node *s* can be infected again after worms have spread via some intermediate nodes: *S*-$N_1$-$N_2$-*S*, *S*-$N_3$-$N_1$-$N_2$-*S*. These two cycles (①③④ and ②③④) start from the initial infectious nodes and end up at

themselves. This self-propagation leads to an infinite increase in the infected probabilities of these initial infectious nodes. In order to eliminate the errors caused by this type of cycle, we define the infected probability of each node to itself as zero in the procedure of the worm's propagation.

Fig. 3.3(c) shows the third type of cycle. $N_1$ can be infected by infectious node *s* directly or via one or more intermediate nodes: *S*-$N_4$-$N_1$-$N_2$-$N_1$, *S*-$N_1$-$N_2$-$N_3$-$N_1$. Two cycles (③and④) begin from the intermediate nodes ($N_1$) and ends up at itself when the worm propagates via some other intermediate nodes ($N_2$, $N_3$). Since two cycles form in the procedure of the worm's spreading but not from the initial infectious nodes, we cannot eliminate the infinite probability cycles by setting the diagonal items in PM to zero. The macroscopic propagation models cannot exclude the errors caused by propagation cycles among the intermediate nodes. Thus, it is desirable to have a mathematical model quantifying the errors and discussing the impact on the worm's propagation.

In order to avoid the errors in non-reinfection worms, we introduce an error calibration vector *E*, as shown in (3.18):

$$\begin{aligned} &E^{(k)} = \left[e_1^{(k)}, e_2^{(k)}, \ldots, e_i^{(k)}, \ldots, e_n^{(k)}\right]^T, \\ &e_i^{(k)} = \sum_{x=1}^{k-1} p_{si}^{(k-x)} p_i^{(x)}, \quad i = 1....n, k \geq 2, \end{aligned} \tag{3.18}$$

where *k* is the current iteration times. $P_{si}^{(k-x)}$ is the propagation probability from node *s* to node *i* by (*k*-*x*) times' iteration. $P_i^{(x)}$ is the propagation probability from node *i* to node *i* by *x* times' iteration. Consequently, in the case of non-reinfection worms, we calibrate $I_{svq}^{(k)}$ to be (3.19):

$$I_{svqe}^{(k)} = I_{svq}^{(k)} - E_{svq}^{(k)} \quad (k \geq 2) \tag{3.19}$$

### 3.3.5 Propagation Ability (*PA*)

In real-world scenarios, attackers expect to control a significant proportion of a network to enable worm propagation. The worm propagation ability (*PA*) is related to the number of peers that the worm can propagate to with high probability and related time delay. In consideration of more than one path for the propagating worm, we adopt an accumulative *I (AI)* to represent the sum of probabilities for the worm propagation between two peers with at most *k* intermediate nodes. It is defined in (3.20):

$$\begin{aligned} AI &= \left[ai_1, ai_2, \ldots, ai_i, \ldots, ai_n\right]^T \\ ai_i &= \frac{\sum_{k=0}^{n-2} \mathrm{Re}(ie_i^{(k)})}{n-1} \quad ai_i \in [0,1) \end{aligned} \tag{3.20}$$

where in function $Re(I^{(k)})$ is used to obtain the real component of $I^{(k)}$, $n$ indicates the number of nodes in the network and $(n-1)$ means the maximum number of intermediate nodes. In the propagation procedure, it is observed that the infected probability gradually decreases when the number of intermediate nodes increases.

Moreover, we define the accumulative time delay *AT* to represent the estimated time delay for the worm propagation between two peers with at most *k* intermediate nodes, as shown in (3.21).

$$\begin{aligned} AT &= \left[at_1, at_2, \ldots, at_i, \ldots, at_n\right]^T \\ at_i &= \frac{2\sum_{k=0}^{n-2} \mathrm{Im}(ie_i^{(k)}) \times \mathrm{Re}(ie_i^{(k)})}{(n-1)\sum_{k=0}^{n-2} \mathrm{Re}(ie_i^{(k)})} \quad at_i \in [0,1] \end{aligned} \tag{3.21}$$

The condition to terminate propagation is when the matrix iteration count reaches *N-2* (*N* nodes in a network). Since *PA* is two-tuples (*AI*, *AT*), in order to evaluate the *PA*, we simply inspect the *AI* and *AT* for each node in the network after an iteration of propagation.

## 3.4 Summary

Most macroscopic models simulate the overall scale of an infected network for each time tick, making them invalid for examining deep inside the propagation procedure among individual nodes. For this reason, this chapter compared the differences between the existing macroscopic and microcosmic worm propagation models and proposed a new microcosmic exploration for modeling worm propagation processes. Firstly, we presented a complex matrix model to construct the propagation of worms from one node to another node. Our model involves three indispensible aspects for propagation: infected state, vulnerability distribution and patch strategy. Through analyzing different scenarios of these three aspects, we can generate a set of optimized patch strategies so that defenders can prevent the worms from spreading using a reasonable and economic approach. In the proposed model, we use three different vectors to represent these key factors: propagation source vector, vulnerable distribution vector and patch strategy vector. We also discussed propagation cycles in the propagation path that result in propagation errors. In order to quantify the errors, the proposed model introduces an error calibration vector and thus, investigates the impact on the worm's propagation. This model adopts propagation ability to evaluate the propagation procedure of worms.

The proposed microcosmic worm propagation model is able to provide a series of recommendations and advice for patch strategies to counter worm propagation. We apply the proposed microcosmic model to observe the propagation of scanning worms through the design of different experiments in Chapter 4. According to the results, our microcosmic model can successfully provide useful information for the proposed problems of where, when and how many nodes we need to patch.

# Chapter 4

# Microcosmic Modeling of the Propagation and Defense Study of Scanning Worms

Scanning worms scan IP addresses to infect vulnerable computers in the network. This chapter applies the proposed microcosmic worm propagation model in Chapter 3 to analyze the propagation procedures of scanning worms, such as Code Red II. The objectives of this chapter are to address three practical aspects of preventing worm propagation: 1) Where do we patch? 2) How many nodes do we need to patch? 3) When do we patch? We implement a series of experiments to evaluate the effects of each major component in the microcosmic model proposed in Chapter 3 and provide a set of optimized and economical patch strategies to prevent scanning worms from spreading. Based on the results drawn from the experiments, for high risk vulnerabilities, it is critical that networks reduce the number of vulnerable nodes to below a certain threshold, e.g., 80% in this analysis. We believe the results can benefit the security industry by allowing them to save significant money in the deployment of their security patching schemes. Moreover, we investigate the mutual impact between the propagation probability and time delay and discuss the overestimation caused by errors in macroscopic models. Through the analysis of the propagation procedure, we observe that the

error is mainly caused by propagation cycles in the propagation path, which are usually ignored by traditional macroscopic models.

## 4.1 Introduction

Each year, in order to prevent worms from spreading effectively, large amounts of money and labor are spent by industry on patching vulnerabilities in operating systems and popular software. Wipro Technologies stated in their 2004 patch management costs report [67], "Annual per-system patching costs on windows: $297.1(clients), $416.2 (Non-Database Servers), $682.1 (Database Servers) and on open source software systems: $343.7 (clients), $479.3 (Non-Database Servers), $1020.4 (Database Servers)." We expect the cost to have been greater in 2010 because of the enormous increase in sophistication and potential for damage caused by worms. Consequently, it is important to provide a set of optimized and economic patch strategies to deal with the problems of where and how many nodes we need to patch.

Security experts routinely uncover software vulnerabilities and then issue software patches and upgrades. Sometimes, however, it may cause inadvertent and possibly detrimental effects. Security researcher Dan Kaminsky uncovered a flaw in the Domain Name System (DNS) and published a series of patches before publicly disclosing the specifics of the vulnerability [68]. By looking at the patch, others were able to reverse engineer the patch, and shortly afterwards code to exploit the newfound weakness had been posted to a website. Some network administrators may have initially been reluctant to patch their systems, fearing that the upgrade itself might cause problems. However, the result is the potential break out of worms before a sufficient number of nodes can be patched. Therefore, we need to quantify an appropriate time for patching vulnerabilities.

In order to understand and possibly address: 1) where to patch; 2) how many nodes we need to patch; 3) when to patch, we characterize the worm propagation through the microcosmic model proposed in Chapter 3. We mainly focus on scanning worms in this chapter, which scan the entire network and explore the vulnerabilities without regard to topological constraints. It is closely related to the logical features of the network rather than the physical structure. Therefore, our proposed approach is suitable for modeling networks that are susceptible to scanning worms.

The objective of the research is to generate a set of optimized patch strategies to minimize the number of infected peers and provide economic benefits to industry by selectively deploying security patches. The major contributions of the chapter are as follows. Firstly, according to the microcosmic model proposed in Chapter 3, we carry out extensive simulation studies of worm propagation and successfully provide useful information for the proposed problems of where, when and how many nodes we need to patch. Secondly, through deploying different scenarios, we can find how propagation source states, vulnerabilities distributions and patch strategies impact on the spreading of worms. Thirdly, we derive a better understanding of dynamic infection procedures in each step of matrix iteration. These procedures include: 1) what is the propagation probability and time delay between each pair of nodes; 2) how does one node infect another node directly; 3) how does one node infect another node through a group of intermediate nodes.

The rest of this chapter is organized as follows. In Section 4.2, we introduce the design of our experiments, including the experiment environment and scanning strategy. In Section 4.3, we evaluate three key factors of the proposed model. Then we analyze the effect of the impact factor and the overestimation in macroscopic models in Section 4.4 and Section 4.5 respectively. In Section 4.6, we discuss open issues. Finally, we conclude this chapter in Section 4.7 with a brief summary.

## 4.2 Design of Experiments

Our implementation is in Visual C++ 2008 SP1 and Matlab 7. The random numbers in our experiments are produced by the C++ TR1 library extensions. Experiments are carried out by a series of simulations: 1) we analyze the effect of the main components in our model including *S*, *V*, *Q*; 2) we analyze the mutual effect from the impact factor $\beta$ between propagation probability and time delay; 3) in this chapter we focus on scanning worms that primarily belong to the non-reinfection class of worms. Thus, we evaluate the errors caused by loops in worm propagation, which are normally ignored by macroscopic propagation models.

Some worms, such as Code Red [2], Code Red II [15, 29-30], and Slammer [12] can propagate without a dependency on the topology. This means that an infectious node is able to infect an arbitrary vulnerable peer. Up to now, many researchers have modeled this type of worm propagation. In our experiments, we choose a typical local preference worm on the basis of Code Red II, as shown in Fig.4.1. The time delay between each pair of nodes follows the Gaussian Distribution N $(0.5, 0.2^2)$.

In practice, there are problems to overcome in the propagation simulation. It often takes a significant amount of time to perform the experiments--72 h in our case on an Intel (R) Core (TM) i7 CPU 2.67-GHz (4 cores) processor to model 10000 nodes--to simulate a single run of matrix iteration for one set of components *S*, *V*, *Q*. To identify trends, many such runs need to be performed and the whole simulation process has to be rerun for any parameter changes. The simulation overhead can be prohibitively high in some cases when the simulated network has a larger scale. This leads to the conclusion that all such experiments are intractable in practice. However, according to our practice and observation, we have found two properties of our model that can be used in addressing the difficulties stated above: 1) our model is

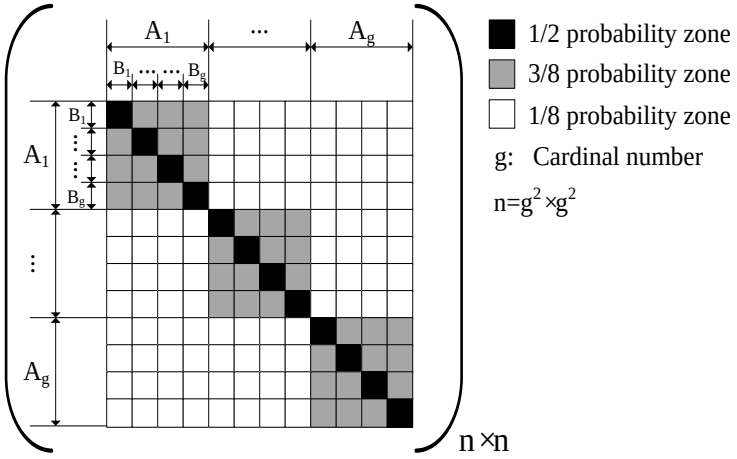



Figure 4.1: Code Red II probability propagation matrix

based on matrix computation (See Formula (3.3) in Chapter 3). Matrix multiplication has a computational cost, however, the matrix operations all run with a polynomial time complexity and can be highly parallelized. Matrix multiplication is the bottleneck in our implementation and is an embarrassingly parallel problem which means each resultant matrix element can be solved concurrently. Thus, the performance of our system will increase significantly with the addition of concurrent computational resources. On a single workstation, we performed the evaluation using 2×4 threads (OpenMP threading Library) to improve the speed of matrix computation. The theoretical speedup is linear in an embarrassingly parallel problem such as matrix multiplication for most realistically sized clusters, which means the computational time is reduced linearly as more computational units are utilized. Industry and research organizations have access to significant computation resources and can mitigate the performance obstacles we have described by employing distributed and high performance computing resources such as clusters and clouds; 2) we analyze the impact of changing the matrix dimensionality used in the experiments and find that a larger dimension will not produce significantly different results. In order to show these results clearly, we choose reasonable network sizes (5000 nodes) and examine them under different scenarios.

Table 4.1 Scenarios for Analysing Propagation Source (*S*)

| ***Scenario*** | ***Description (refer to Fig. 4.1)*** | ***Practical Meaning*** |
|---|---|---|
| 1 | IP address range $A_1B_1$ has increasing number of initial infectious nodes. | Analyzing the impact of the number of initial infectious sources on the propagation probability in an IP address range such as a specific region. |
| 2 | Increasing number of IP address ranges $A_1B_x$ ($x \in [1, g]$) have an initial infectious node. | Analyzing the impact of different geographic distribution of initial infectious sources on the propagation probability. |
| 3 | Increasing number of IP address ranges $A_xB_1$ ($x \in [1, g]$) have an initial infectious node. | |
| 4 | IP address ranges $A_xB_y$ ($x, y \in [1, g]$) have a different number of initial infectious nodes. | Analyzing worm propagation when different regions have a different density of initial infectious source. |

## 4.3 Effect of Three Key Factors

In this section, we evaluate the effects of three significant factors for scanning worms: infected state, vulnerability distribution and patch strategy according to different scenarios. Then, based on the results, we derive a series of recommendations and provide advice for patch strategies.

### 4.3.1 Effect of the Propagation Source Vector

In this subsection, we assume all nodes in the network are vulnerable and no nodes have been patched. According to the Symantec Internet Security Threat Report [59], global malicious activities are not evenly distributed in different ranges of IP addresses. Consequently, we arrange a group of scenarios with practical meaning in Table 4.1 to describe the different origins of worms. The results are represented by the mean value of propagation ability ($E$(AI)), the variation of propagation ability ($D$(AI)), the mean value of propagation time delay ($E$(AT)) and the variation of propagation time delay ($D$(AT)). In order

Table 4.2 Results from Different Scenarios of Propagation Source (*S*)

| ***Scenario*** | ***Infectious Node*** | | ***Propagation Probability*** | | ***Time Delay*** | |
|---|---|---|---|---|---|---|
| | ***Quantity*** | ***Scenario Setting*** | ***E(AI)*** | ***D(AI)*** ($\times 10^{-4}$) | ***E(AT)*** ($\times i$) | ***D(AT)*** ($\times 10^{-2}$) |
| 1 | 1% | $A_1B_1$ has 1% initial infectious nodes | 0.0124 | 0.0729 | 0.5201 | 0.3758 |
| | 2% | $A_1B_1$ has 2% initial infectious nodes | 0.0124 | 0.0729 | 0.2605 | 0.0941 |
| | 3% | $A_1B_1$ has 3% initial infectious nodes | 0.0124 | 0.0729 | 0.1735 | 0.0416 |
| 2 | 2% | $A_1B_1$ and $A_1B_2$ have 1% initial infectious nodes respectively | 0.0124 | 0.0723 | 0.2605 | 0.0938 |
| | 3% | $A_1B_1$, $A_1B_2$ and $A_1B_3$ have 1% initial infectious nodes respectively | 0.0124 | 0.0704 | 0.1734 | 0.0418 |
| 3 | 2% | $A_1B_1$ and $A_2B_1$ have 1% initial infectious nodes respectively | 0.0124 | 0.0320 | 0.2576 | 0.0659 |
| | 3% | $A_1B_1$, $A_2B_1$ and $A_3B_1$ have 1% initial infectious nodes respectively | 0.0124 | 0.0183 | 0.1706 | 0.0203 |
| | 4% | $A_xB_1$ ($x \in [1, 4]$) have 1% initial infectious nodes respectively | 0.0124 | 0.0115 | 0.1275 | 0.0079 |
| 4 | 3% | $A_1B_1$ has 2% infectious nodes and $A_1B_2$ has 1% infectious nodes | 0.0124 | 0.0724 | 0.1732 | 0.0414 |
| | 3%* | $A_1B_1$ has 2% infectious nodes and $A_2B_1$ has 1% infectious nodes | 0.0124 | 0.0365 | 0.1722 | 0.0306 |

to describe the differences of each parameter clearly, we cut the first 81 nodes to make figures for some experiments.

#### *4.3.1.1 Scenario 1*

***Preparation*:**

We deploy 1% to 3% infectious nodes in $A_1B_1$ of PM (See Fig.4.1). Based on different propagation probabilities, the entire IP space is divided into three ranges:

- $R_1$: $A_1B_1$
- $R_2$: $A_1B_2 \rightarrow A_1B_g$
- $R_3$: $A_2 \rightarrow A_g$

***Result*:**

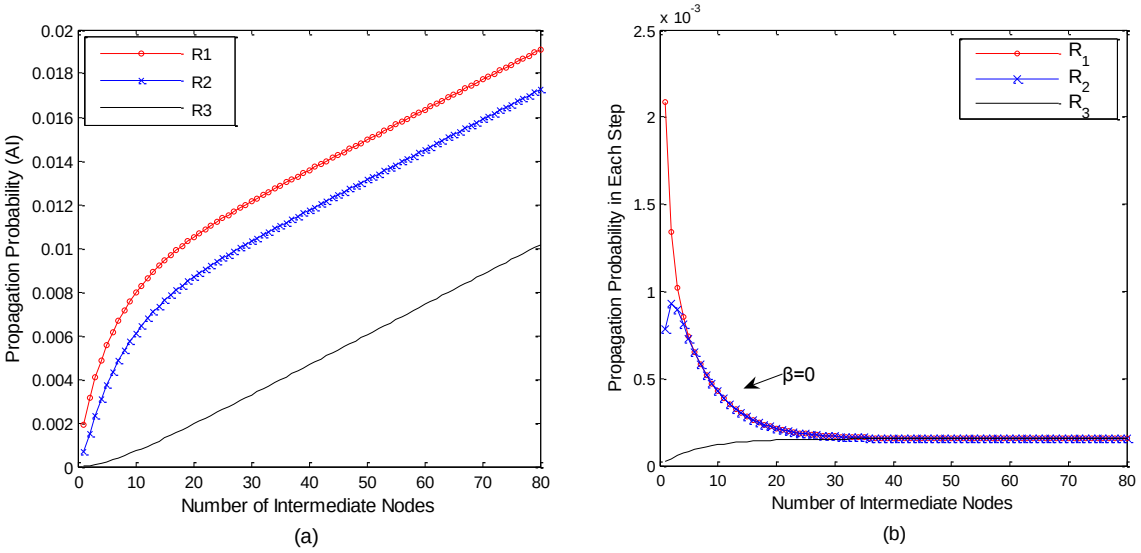



Figure 4.2: Propagation probability in scenario 1 (the first 81 nodes in 5000 nodes)

The result is listed in Table 4.2 Scenario 1. We find that the number of initial infectious nodes have no impact on *E*(AI) and *D*(AI). As shown in Fig. 4.2, AI in different IP ranges $R_1$, $R_2$ and $R_3$ are overlapped respectively when the number of initial infectious nodes increases. In Fig. 4.2(a), AI deviates in different IP ranges during the propagation procedure: $AI(R_1)>AI(R_2)>>AI(R_3)$. In Fig. 4.2(b), the difference of AI deviates in different IP ranges. Within the first 20 iterations, $R_1$ and $R_2$ decline rapidly, while $R_3$ slightly increases. Afterwards, the difference of AI tends to be stable.

In Table 4.2, the result of time delay reflects temporal properties of the worm propagation in this scenario; an increasing number of initial infectious nodes results in a decrease in *E*(AT) and *D*(AT). Fig.4.3 shows the estimated time delay AT in different IP ranges when the number of initial infectious nodes increases. During the first nearly 40 iterations, $AT(R_1)>AT(R_2)>AT(R_3)$; afterwards, AT in $R_3$ goes up quickly: $AT(R_3)>AT(R_2)>AT(R_1)$.

***Analysis*:**

Although the number of initial infectious nodes is increasing, their effects are limited in the same IP ranges, which leads to the overall propagation probabilities are not improved. Therefore the value of *E*(AI) and *D*(AI) stay the same.

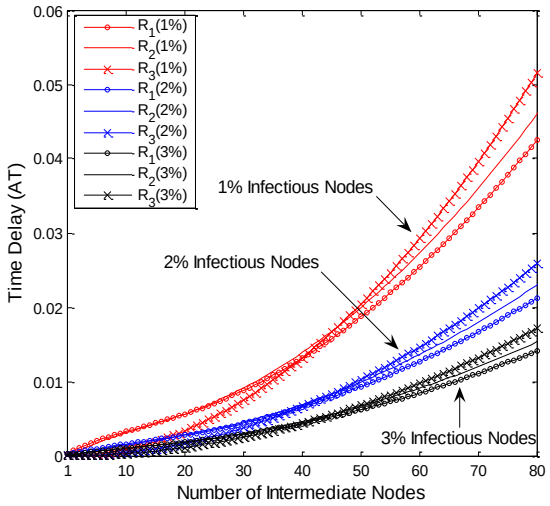



Figure 4.3: Propagation time delay in scenario 1 (the first 81 nodes in 5000 nodes)

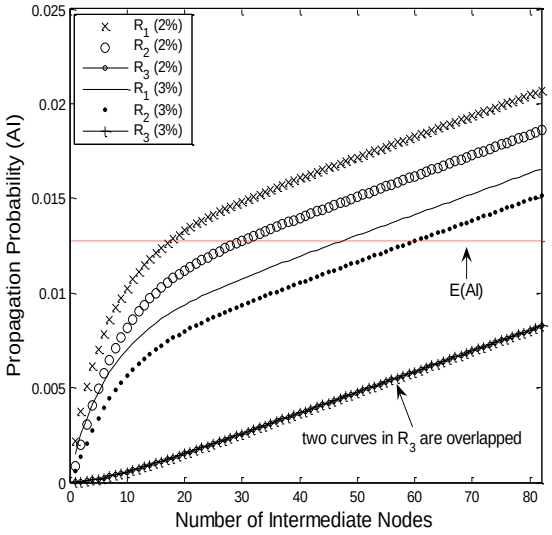



Figure 4.4: Propagation probability in scenario 2 (the first 81 nodes in 5000 nodes)

When more infectious nodes are involved, the *E*(AT) obviously decreases as the average time for searching the targets is reduced. Meanwhile, a decline of *D*(AT) indicates that an increase in the number of initial infected nodes can accelerate the propagation speed to all nodes in the network since the time delay is close to the *E*(AT). In the early propagation stage, the infected nodes are mainly in IP ranges $R_1$ and $R_2$. Thus, AT is dominated by the nodes with greater propagation probability in $R_1$ and $R_2$. Afterwards, when the number of infected nodes in $R_3$ increases, the nodes in $R_3$ have greater contribution to AT.

#### *4.3.1.2 Scenario 2*

***Preparation*:**

We deploy 2% and 3% infectious nodes in PM (See Fig.4.1). Based on different propagation probabilities, the entire IP space is divided into three ranges:

- $R_1$: $A_1B_1 \rightarrow A_1B_2$ (2% infectious nodes)

  $A_1B_1 \rightarrow A_1B_3$ (3% infectious nodes)

- $R_2$: $A_1B_3 \rightarrow A_1B_g$ (2% infectious nodes)

  $A_1B_4 \rightarrow A_1B_g$ (3% infectious nodes)

- $R_3$: $A_2 \rightarrow A_g$

***Result*:**

The result is listed in Table 4.2 Scenario 2. As shown in Fig. 4.4, the basic tendency of the curves is similar to scenario 1. However, more infectious nodes (from 2% to 3%) in the network result in a decrease in AI of $R_1$ and $R_2$. Additionally, we find that the number of initial infectious nodes has no impact on AI in $R_3$.

In Table 4.2, temporal properties of time delay in scenario 2 stay the same with scenario 1. As shown in Fig.4.5, the value of AT decreases when the number of initial infectious nodes increases.

***Analysis*:**

We analyze the decrease of AI in $R_1$ and $R_2$ when there are more initial infectious nodes distributed in adjacent IP ranges of the network. The reason for this is that when a new infectious node in $A_1B_3$ is involved, compared with two infectious nodes case, the AI in $A_1B_3$ will increase. However, the sum of all probabilities is equal to one, which means an increase

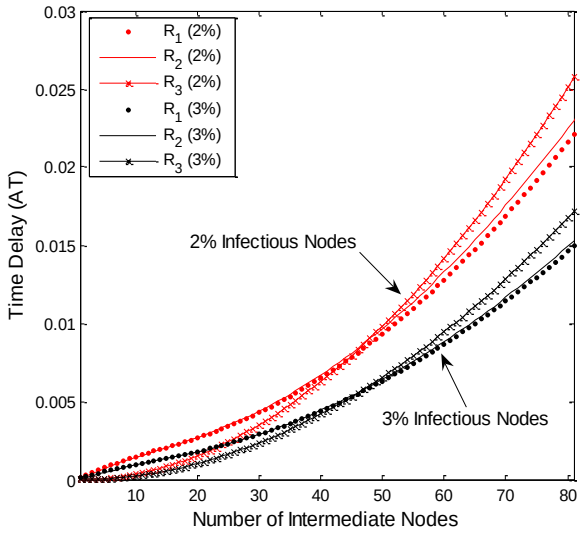



Figure 4.5: Propagation time delay in scenario 2 (the first 81 nodes in 5000 nodes)

of AI in $A_1B_3$ results in a mathematical decrease of AI in other infectious ranges such as $A_1B_1$-$A_1B_2$.

Similar to scenario 1, $E$(AI) stays the same (0.0124), and only a small decrease $D$(AI) (from $0.0723\times10^{-4}$ to $0.0704\times10^{-4}$) indicates more nodes in the network have higher probabilities of being infected. Additionally, scenario 2 has the same acceleration of propagation time as scenario 1.

#### *4.3.1.3 Scenario 3*

***Preparation*:**

We deploy 2% to 4% infectious nodes in PM (See Fig. 4.1). Based on different propagation probabilities, the entire IP space is divided into three ranges:

- $R_1$: $A_1B_1+A_2B_1$ (2% infectious nodes)

  $A_1B_1+A_2B_1+A_3B_1$ (3% infectious nodes)

  $A_1B_1+A_2B_1+A_3B_1+A_4B_1$ (4% infectious nodes)

- $R_2$: $\{A_xB_2\rightarrow A_xB_g\}_{x=1,2}$ (2% infectious nodes)

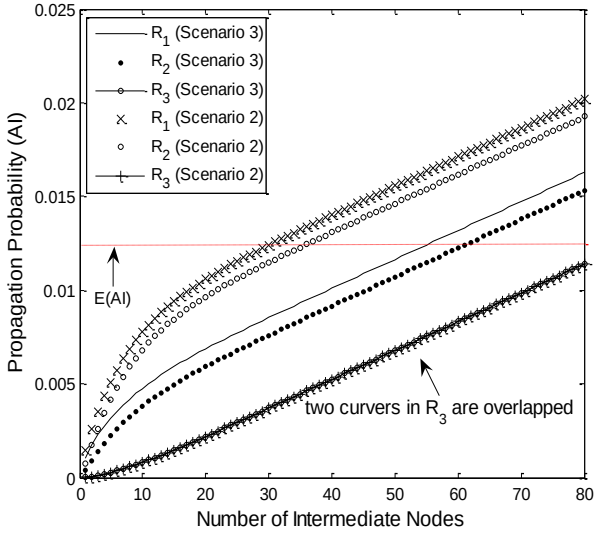



Figure 4.6: Propagation probability in scenario 2 and scenario 3
(the first 81 nodes in 5000 nodes)

$\{A_xB_2 \rightarrow A_xB_g\}_{x=1,2,3}$ (3% infectious nodes)

$\{A_xB_2 \rightarrow A_xB_g\}_{x=1,2,3,4}$ (4% infectious nodes)

- $R_3$: $A_3 \rightarrow A_g$(2% infectious nodes)

  $A_4 \rightarrow A_g$(3% infectious nodes)

  $A_5 \rightarrow A_g$(4% infectious nodes)

***Result*:**

We use 2% infectious nodes case to compare with scenario 2. In Fig. 4.6, when the infectious nodes are scattered in the network, the AI of $R_1$ and $R_2$ decreases. AI of $R_3$ stays the same.

In Table 4.2, temporal properties of time delay in scenario 3 stay the same with scenario 2. In Fig. 4.7, AT of scenario 2 is almost the same with AT of scenario 3 in IP ranges $R_1$ and $R_2$.

***Analysis*:**

We analyze the decrease of AI in $R_1$ and $R_2$ when the initial infectious nodes are scattered in different IP ranges of the network. The reason is that when infectious nodes are deployed

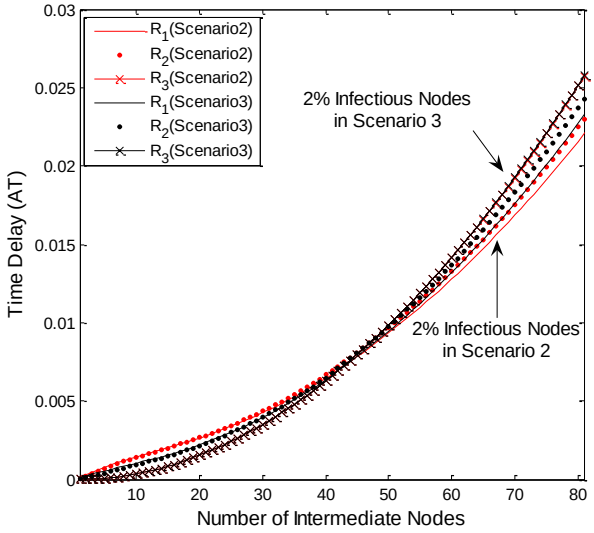



Figure 4.7: Propagation time delay (scenario 2 vs. scenario 3)
(the first 81 nodes in 5000 nodes)

loosely, more nodes have a higher probability of being infected. Similar to the exceptional decrease of AI in scenario 2, an increase of AI in $A_2B_1 \rightarrow A_2B_g$ results in a mathematical decrease of AI in other infectious ranges.

Additionally, scenario 3 has the same acceleration of propagation time as scenario 2 and 1.

#### *4.3.1.4 Scenario 4*

***Preparation*:**

We deploy 3% infectious nodes in PM (See Fig. 4.1). Based on different propagation probabilities, the entire IP space is divided into several ranges:

- $R_1$: $A_1B_1$
- $R_2$: $A_1B_2$ (3% infectious nodes)

  $A_2B_1$ (3%* infectious nodes)
- $R_3$: $A_1B_3 \rightarrow A_1B_g$ (3% infectious nodes)

  $A_1B_2 \rightarrow A_1B_g$ (3%* infectious nodes)

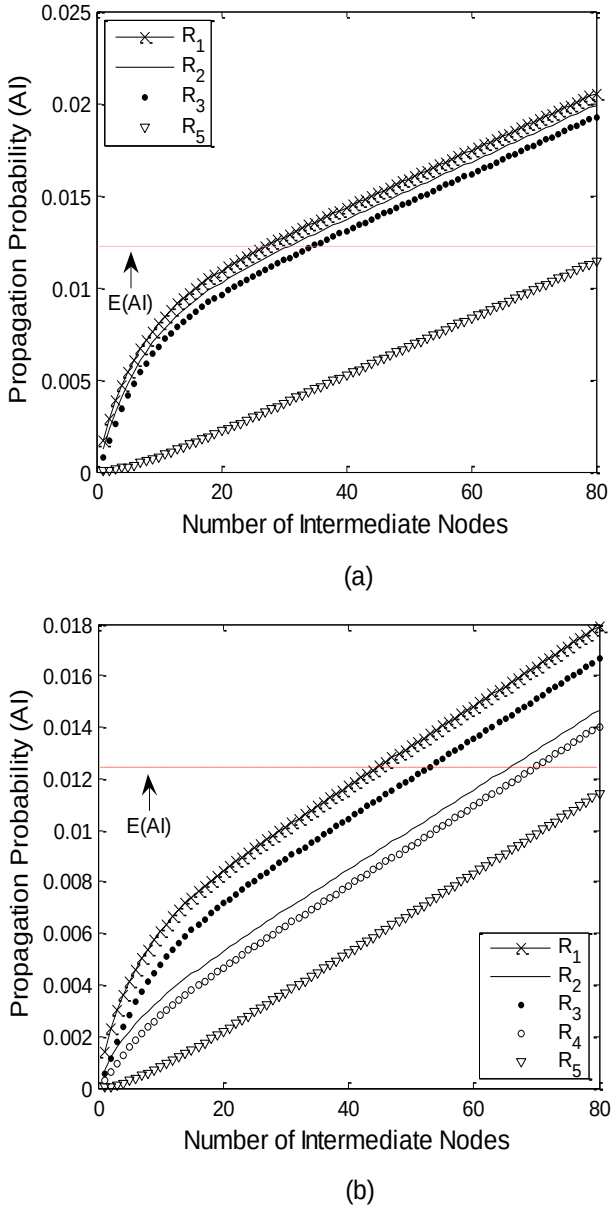



Figure 4.8: Propagation probability in scenario 4 (the first 81 nodes in 5000 nodes)

- $R_4$: $A_2B_2 \rightarrow A_2B_g$ (3%* infectious nodes)
- $R_5$: $A_3 \rightarrow A_g$

***Result*:**

The result is listed in Table 4.2 Scenario 4. In Fig. 4.8(a), two infectious nodes are in $A_1B_1$, and another infectious node is in $A_1B_2$. The result shows four IP address ranges have different

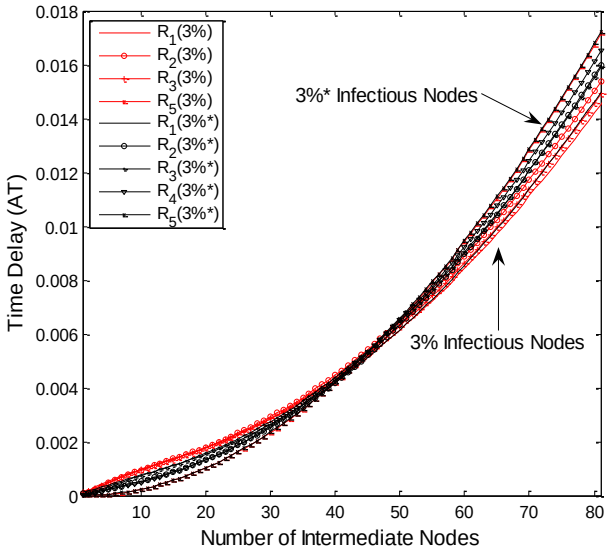



Figure 4.9: Propagation time delay in scenario 4 (the first 81 nodes in 5000 nodes)

AI. In Fig. 4.8(b), two infectious nodes are in $A_1B_1$, and another infectious node is in $A_2B_1$. The result shows five IP address ranges have different AI.

In Table 4.2, scenario setting in Fig. 4.8(a) spends slightly more time infecting the nodes in the network than Fig. 4.8(b) (0.1732 compared with 0.1722). In Fig. 4.9, the value of AT is almost the same when the same proportion of initial infectious nodes are deployed in different IP ranges.

***Analysis*:**

We analyze the reason of four and five different ranges of AI. An infectious node has larger effect on its own and adjacent IP ranges. A high density of initial infectious nodes has greater effect on its own and adjacent IP ranges than other IP ranges with low density. Therefore, in Fig. 4.8(a), $R_1$(2% initial infectious nodes) has higher AI than $R_2$ (1% initial infectious nodes). In Fig. 4.8(b), R3 that is adjacent to $R_1$ (2% initial infectious nodes) has higher AI than $R_4$ that is adjacent to $R_2$ (1% initial infectious nodes).

#### *4.3.1.5 Conclusion of Propagation Source Effect*

We draw conclusions on the practical meaning from different scenarios of the propagation source.

- In scenario 1, an increasing number of initial infectious nodes in a specific region has no impact on propagation probability (AI) in the entire network. However, it does accelerate the speed of worm propagation considerably.
- Within a certain (20 in scenario 1) number of intermediate nodes, the vulnerable nodes in adjacent IP ranges of an infectious source have a greater probability of being infected.
- In scenario 2 and 3, different geographic distribution of initial infectious nodes has no impact on the overall AI. However, when initial infectious nodes are more scattered in the network, they can infect more vulnerable nodes in the adjacent IP address ranges and accelerate the speed of worm propagation considerably in the network.
- In scenario 4, a high density of initial infectious nodes can infect more vulnerable nodes, which are mainly in adjacent IP address ranges of an infectious source.

#### *4.3.1.6 Inspiration for Developing the Patch Strategy*

The experiments on the propagation source vector ($S$) are mainly used to estimate where we need to patch.

- ***Where***: According to the conclusion in this subsection, the best position for patching are similar or adjacent net blocks to the propagation source. In the real world, however, it is impractical to locate this position since the initial infectious nodes may be scattered and it is difficult to foresee the original propagation sources. On the basis of the conclusion from scenario 4, the IP ranges with a high density of vulnerable

nodes are essential areas in lieu of adjacent IP ranges of a propagation source for patching, since denser ranges have a greater possibility to be chosen as initial infectious sources. This may warrant collaboration across administrative boundaries when adjacent net blocks are not controlled by the same authority. It may be advantageous for network administrators to have a prior relationship with adjacent network owners to work together in threat intelligence and help prevent worm outbreaks and establish patch priorities in their own networks

- ***How many***: The number of nodes that require patching is closely related to the different vulnerability distributions in the network. We will discuss this in the conclusion of the next subsection.
- ***When***: Here, we will consider the estimated time of worm propagation in scanning worms. This is closely related to the propagation probability in the target IP ranges, but is unrelated to the geographic distribution of the propagation sources. In our experiments, when the percentage of the initial infectious nodes was from 1% to 4%, the range of propagation time delay was from 0.13$i$ to 0.52$i$. $i$ is the scanning time of the entire IP address space.

### 4.3.2 Effect of the Vulnerable Distribution Vector

In this subsection, we assume that not all nodes are vulnerable and that no nodes have been patched. Symantec examines the types of worms causing potential infections in each region [59]. The increasing regionalization of vulnerabilities is observed from one area to the next when vulnerabilities concern certain languages or localized events. Information about the geographic distribution of vulnerabilities can help network administrators improve their security efforts. Consequently, we arrange a group of scenarios with practical meaning in

Table 4.3 Scenarios for Analyzing Vulnerability Distribution (*V*)

| ***Scenario*** | ***Description*** | ***Practical Meaning*** |
|---|---|---|
| 1 | Increasing percentage of vulnerable nodes and the vulnerabilities follow uniform distribution. | Analyzing worm propagation when most of the nodes are vulnerable without the difference of geographic distribution. |
| 2 | Increasing percentage of vulnerable nodes and the vulnerabilities follow a Gaussian distribution. Initial infectious nodes are deployed in an IP address range that is rich in vulnerable nodes. | Analyzing the impact of different geographic distribution of vulnerabilities on worm propagation.<br><br>Analyzing the impact of different deployment of a propagation source under different distribution of vulnerabilities. |
| 3 | Increasing percentage of vulnerable nodes and the vulnerabilities follow a Gaussian distribution. Attackers deploy initial infectious nodes in sparse vulnerabilities ranges | |

Table 4.3 to describe the different distributions of vulnerabilities. The results are represented by the mean value of propagation ability ($E$(AI)), the variation of propagation ability ($D$(AI)), the mean value of propagation time delay ($E$(AT)) and the variation of propagation time delay ($D$(AT)).

#### *4.3.2.1 Scenario 1*

***Preparation*:**

In Scenario 1, we assume a vulnerability rate from 5% to 100% and its distribution follows uniform distribution. We fix the initial infectious nodes to 1.

***Result*:**

The result is described in Fig. 4.10. When the vulnerability rate is less than 80%, $E$(AI), $D$(AI), $E$(AT) and $D$(AT) remain at a low level. The change point is when the vulnerability rate is 80%. The steady AI occurs when the vulnerability rate is lower than 70%.

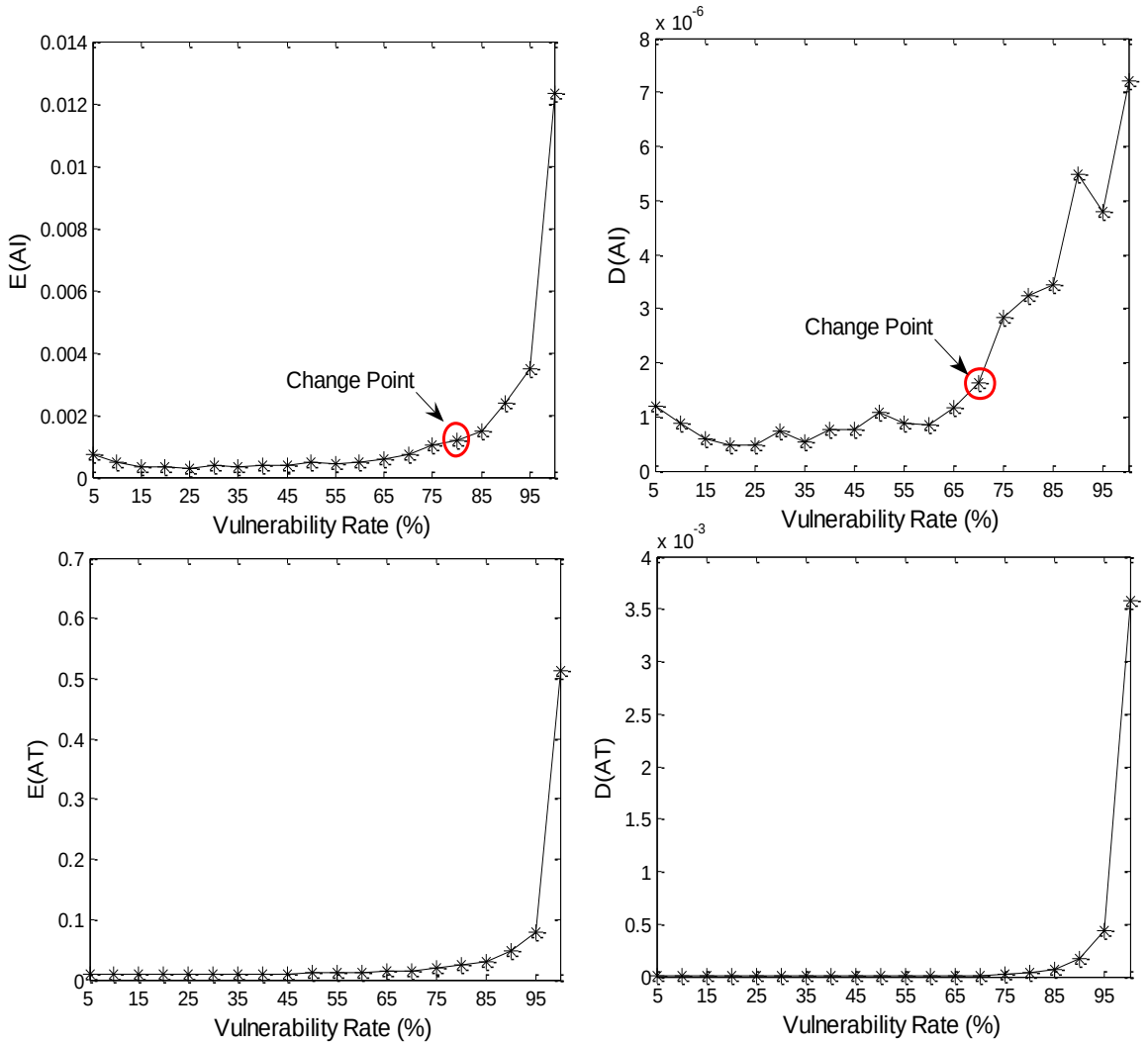



Figure 4.10: Vulnerability in Uniform distribution (scenario 1)

***Analysis*:**

We analyze the reason of the change point at 80%. Drawing from the conclusions of Section 4.3.1, the nodes in adjacent IP ranges of the propagation origins have greater propagation probability to be infected. When the vulnerability follows a Uniform distribution, the coverage rate of vulnerable nodes in adjacent IP ranges of the propagation origins is small if the entire vulnerability rate is not large enough. Therefore, when the vulnerability rate is less than 80%, seldom nodes in adjacent IP ranges are involved in the propagation and the nodes in non-adjacent ranges dominate the value of AI. When the vulnerability rate reaches 80% or more, more vulnerable nodes in adjacent IP ranges may be involved in the propagation, which lead to the $E$(AI) and $D$(AI) increase in Fig. 4.10.

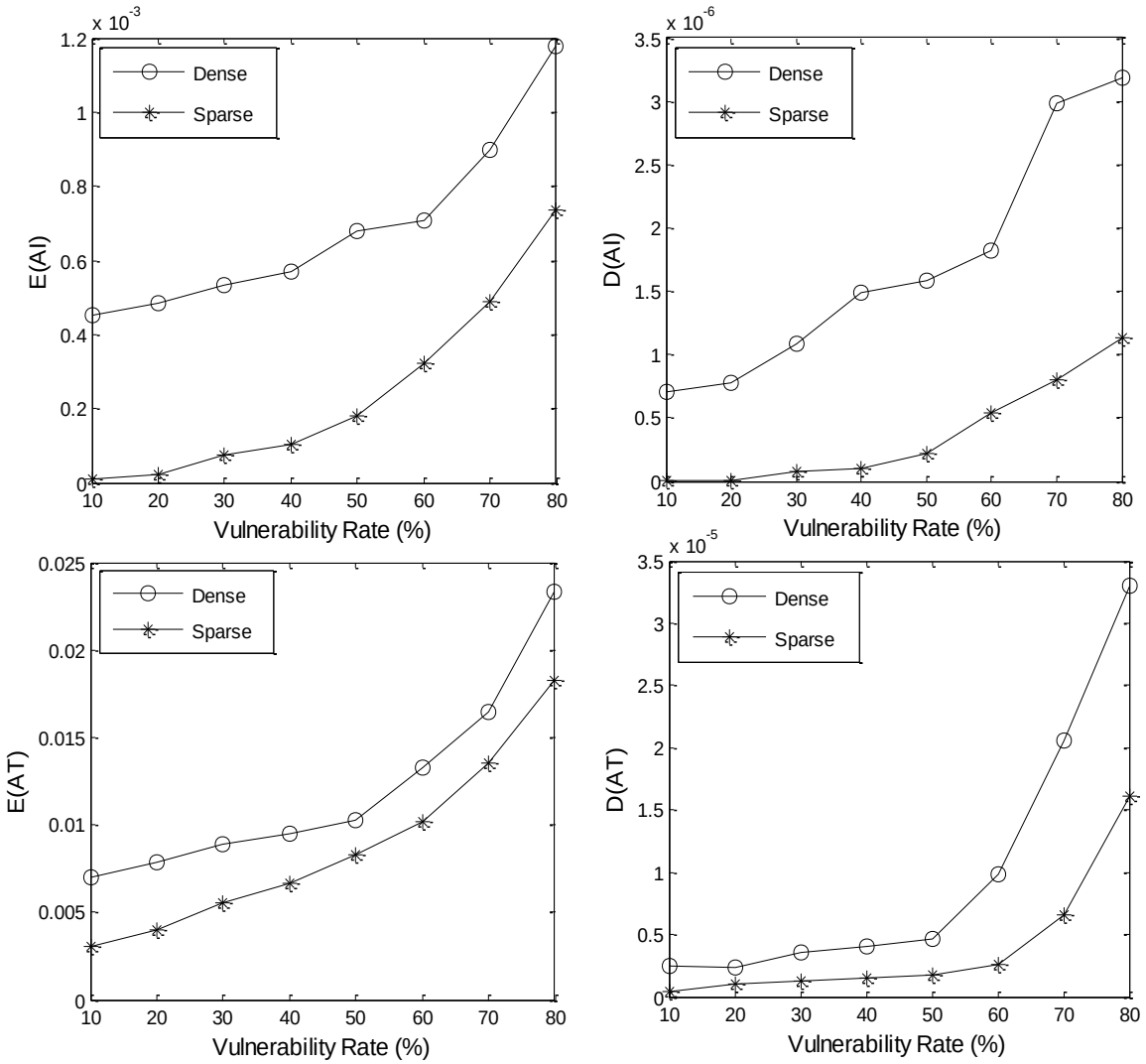



Figure 4.11: Vulnerability in Gaussian distribution (scenario 2 & 3)

When the vulnerability rate is more than 80%, the vulnerable nodes have a large probability of being infected. Thus, an increasing size of infected nodes in the network results in increasing time expenditure for overall worm propagation.

### *4.3.2.2 Scenario 2 and Scenario 3*

***Preparation*:**

In Scenario 2 and 3, we investigate the impact of different geographic distributions of vulnerabilities on worm propagation. We also observe the impact of different deployments of the propagation source under different distributions of vulnerabilities. Therefore, we assume

vulnerabilities follow a Gaussian distribution from N(1024, $102^2$) (10% vulnerability rate) to N(1024, $819^2$) (80% vulnerability rate). We deploy one initial infectious node in vulnerability dense or sparse IP ranges.

***Result*:**

The result is described in Fig. 4.11. When more nodes in the network are vulnerable, $E$(AI) and $D$(AI) gradually increase in different deployments of the initial infectious node. Obviously, if one initial infectious node is in vulnerability dense IP ranges, $E$(AI) and $D$(AI) are larger.

From Fig. 4.11, $E$(AT) and $D$(AT) have similar results to $E$(AI) and $D$(AI).

***Analysis*:**

More nodes in the network are infected when the vulnerability rate increases, which leads to $E$(AI) smoothly increasing. Since the vulnerabilities follow a Gaussian distribution, there are more vulnerable nodes in some specific IP ranges. If the initial infectious nodes are deployed in vulnerability dense IP ranges, the vulnerable nodes in adjacent IP ranges of the propagation origins are quickly infected, which contributes to $E$(AI). This is a reason why $E$(AI) and $D$(AI) are larger when the initial infectious nodes are deployed in vulnerability dense IP ranges.

Similar to scenario 1, an increasing size of infected nodes in the network results in increasing time expenditure for overall worm propagation.

#### *4.3.2.3 Inspiration of the Vulnerable Distribution Effect*

The experiments on the vulnerable distribution vector ($V$) are mainly used to estimate how many nodes we need to patch.

Table 4.4 Scenarios for Analyzing Patching Strategy (*Q*)

| ***Scenario*** | ***Description*** | ***Practical Meaning*** |
|---|---|---|
| 1 | Increasing percentage of patching nodes when vulnerabilities follow Uniform distribution. | Analyzing the effect of patch strategy when most nodes are vulnerable without the difference of geographic distribution. |
| 2 | Increasing percentage of patching nodes when vulnerabilities follow Gaussian distribution. | Analyzing the effect of patch strategy when distribution of vulnerabilities depends on geographic region. |

- ***Where***: If the threat is from a localized worm that exploits vulnerabilities in a specific region, it is of greater value to patch in the areas with a high density of vulnerabilities since the propagation is accelerated when more nodes are vulnerable.
- ***How many***: If the worm propagation is independent of the geographic region, the worm can infect a large number of nodes when the vulnerability rate is more than 80%. Making sure the vulnerability rate is lower than 80% can prevent the worm from propagating effectively. When the vulnerability rate is lower than 70%, the propagation probability remains stable and is significantly lower. A recommended patch strategy is to ensure the vulnerability rate is lower than 70%.

### 4.3.3 Effect of the Patch Strategy Vector

A large amount of money and labor are spent on patching the vulnerabilities each year. In order to reduce the cost, we focus on finding the most economic tactics for corporations to patch their software vulnerabilities. In this subsection, we analyze the effect of patch strategy vector *Q*, which is used to eliminate the vulnerabilities in the vector *V*. Two scenarios are listed in Table 4.4. The results are represented by the mean value of propagation ability

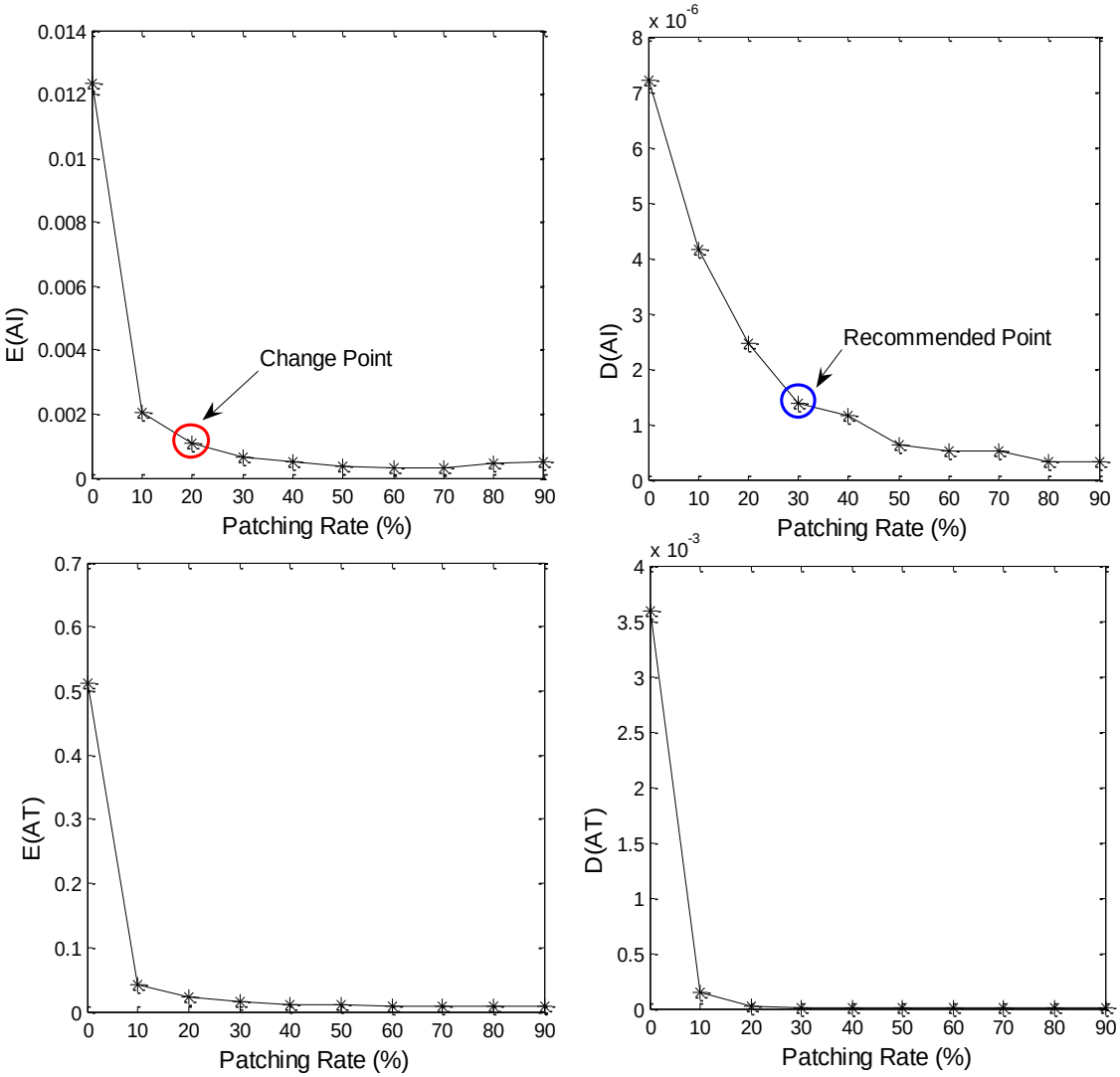



Figure 4.12: Patch strategy (scenario 1)

(*E*(AI)), the variation of propagation ability (*D*(AI)), the mean value of propagation time delay (*E*(AT)) and the variation of propagation time delay (*D*(AT)).

#### *4.3.3.1 Scenario 1*

***Preparation*:**

The intention of patching is to decrease the number of potentially vulnerable nodes. When the patching rate increases, the vulnerability rate decreases. Initially, we assume all nodes are vulnerable and fix one initial infectious node in the network.

***Result*:**

From Fig. 4.12, when the patching rate is higher than 20%, there is no obvious change in *E*(AI). When the patching rate is higher than 30%, *D*(AI) becomes steady. The change points of *E*(AT) and *D*(AT) are at a 10% patching rate.

***Analysis*:**

Once the patching rate reaches 20%, there are no obvious outcomes for more patching. Moreover, the outcomes of the patching strategy become steady when the patching rate is more than 30%.

#### *4.3.3.2 Scenario 2 and Scenario 3*

***Preparation*:**

When vulnerabilities depend on geographic region, some specific IP ranges have more vulnerable nodes. Therefore, we arrange vulnerability to follow a Gaussian distribution. We assume the vulnerability rate is 50% or 80% and fix one initial infectious node in the network. The patching rate varies between 5% to 40%.

***Result*:**

From Fig. 4.13, when the patching rate increases from 5% to 40%, *E*(AI) and *D*(AI) in vulnerable dense IP ranges decrease. The increasing patching rate has a greater effect on *E*(AI) and *D*(AT) with 80% vulnerability rate. Additionally, *E*(AT) and *D*(AT) have a similar tendency.

***Analysis*:**

The objective of this scenario is to investigate the impact of the patching rate on the specific vulnerable dense region. When the specific region has more vulnerable nodes, the patch strategy has more effect.

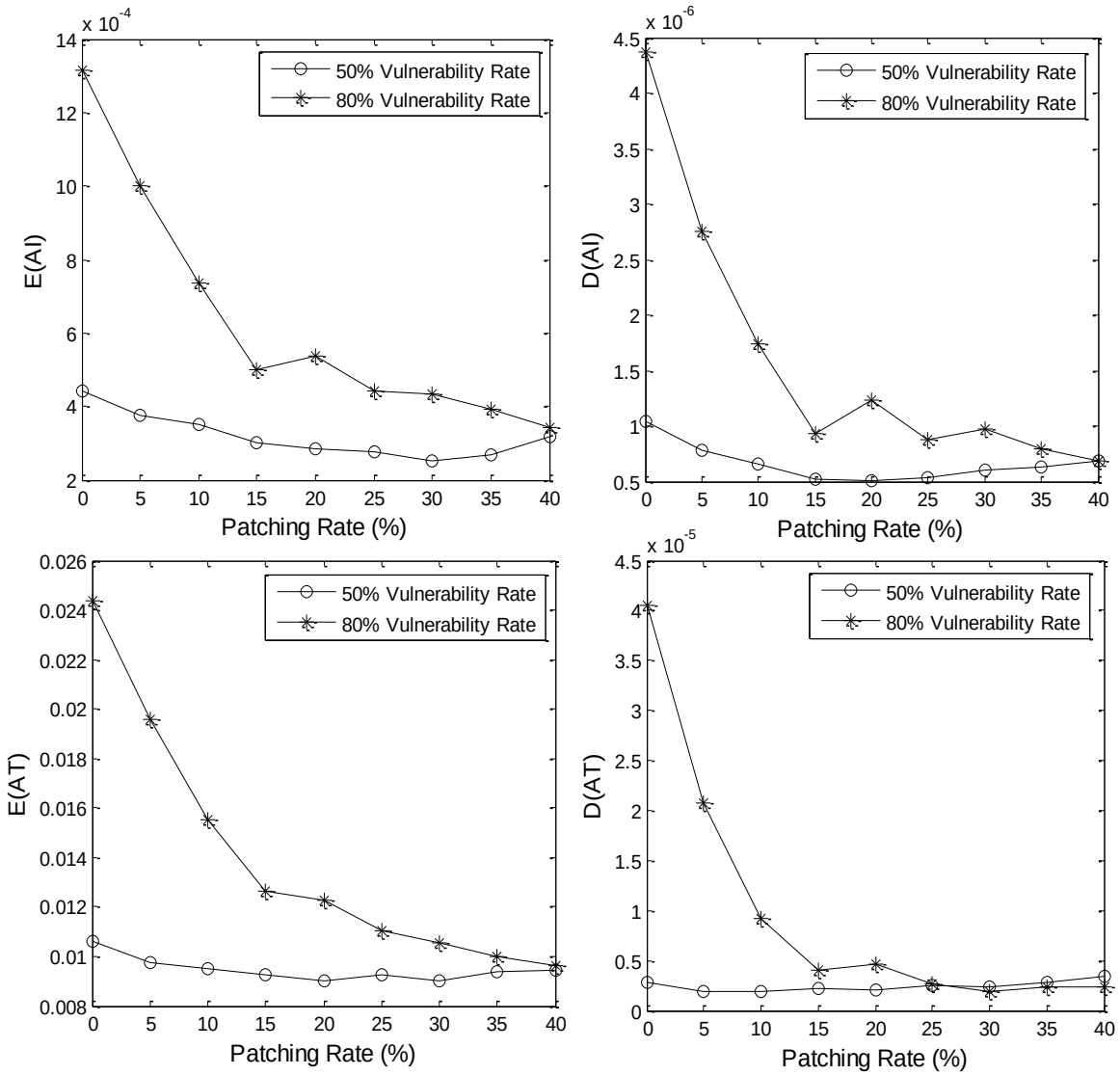



Figure 4.13: Patch strategy (scenario 2)

### *4.3.3.3 Conclusion of the Patch Strategy Effect*

The experiments on the patch strategy vector ($Q$) are mainly used to estimate when we need to patch. In accordance with the conclusions regarding $S$ and $V$, we can summarize the patch strategies.

- ***Where***: If the propagation sources can be predicted, the best strategy is to patch nodes that have the ***same class IP address*** as the infectious sources. However, in real-world scenarios, the propagation sources are hard to locate. In these situations, the IP ranges

with a ***high density*** of vulnerable nodes are essential areas for patching because more nodes are infected in these specific regions.

- ***How many***: The most economic patching rate is ***20%***, however we recommend a ***30%*** patching rate because the outcome of this patch strategy is more stable.
- ***When*****:** "When do we patch?" is a complicated problem when considering global recommendations because it involves many social factors, such as how widely used is the target software or the size of company? However, companies employing vulnerability management services can be given actionable recommendations for when it is critical to patch. For high risk vulnerabilities, it is critical that networks reduce the number of vulnerable nodes to below 80%. Another actionable result is when to disclose information on the vulnerabilities. Most corporations such as Microsoft issue vulnerability patches for software products with some specific information on the nature of vulnerabilities. This ensures that users are aware of the reason and necessity for deploying the patches. However, this information may be utilized by hackers to develop exploits for the vulnerabilities. Therefore, increased disclosure of specific vulnerabilities could possibly be delayed until the patching rate reaches ***at least*** 20%. Otherwise, the worms that target these vulnerabilities can propagate quickly to infect a large proportion of the network.

In the proposed model, the propagation source vector (*S*) and the vulnerable distribution vector (*V*) describe the distribution of initial infectious nodes and the distribution of vulnerable nodes in the network respectively. The patching strategy vector (*Q*) reflects a special deployment of patching nodes. The propagation scale and the spreading speed depend on different deployment of *S*, *V* and *Q*. Through the analysis of propagation probability (*AI*) according to different scenarios of *S*, V, *Q*, we can estimate the best position for patching, the most economic patching rate and the appropriate time for patching.

## 4.4 Effect of the Impact Factor $\beta$

The impact factor $\beta$ reflects the impact of propagation time delay on the propagation probability. We introduced this parameter because the propagation time delay is caused by two factors: the worm's infection strategy and the network infrastructure information such as bandwidth. In 2001, Code Red v1 [29] used a static seed for its random number generator and thus generated identical lists of IP addresses on each infected machine. The first version of the worm spread slowly, because each infected machine began to spread the worm by probing machines that were either infected or impregnable. Then, it was improved in Code Red v2 [29] through generating a random seed variant. This second version shared almost all of its code with the first version, but spread much more rapidly. Each node with an individual IP address may be scanned within a much shorter period of time and consequently the probability of each node to become infected is credibly increased. Therefore, a worm's infection strategy has a significant effect on the spreading time. On the other hand, in 2002, the Sapphire worm [69] randomly selected IP addresses to spread and reached its peak scanning rate of over 55 million scans per second across the Internet in under 3 minutes, but in later stages the rate of growth slowed because networks became saturated with its scans and there was not enough bandwidth to allow the worm to operate unhindered. It is therefore clear that a network environment with more bandwidth will accelerate the infection.

Since we do not know the exact value of $\beta$ for propagation in real worms, we assume $\beta$ is equal to zero, which indicates that the propagation probability cannot be affected by temporal properties in our previous simulations. However, in order to see how the impact factor $\beta$ affects the propagation probability in the worm spreading procedure, we compare the changes of AI with two different $\beta$ by assuming $\beta_1=0.25\times10^{-6}$, $\beta_2=0.5\times10^{-6}$.

***Preparation*:**

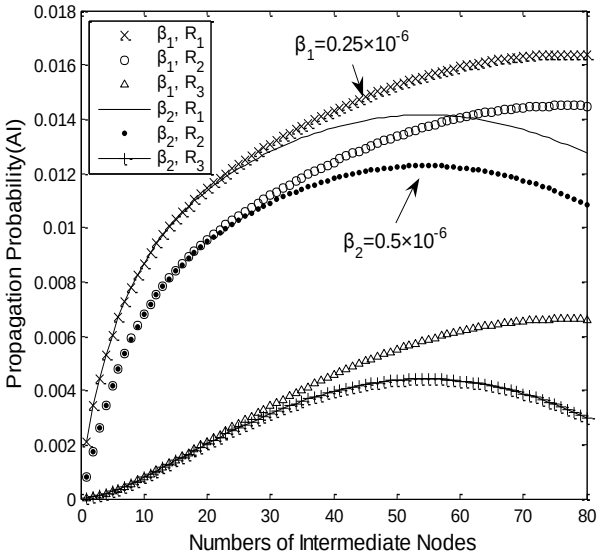



Figure 4.14: Effect of impact factor $\beta$ on worm propagation
(the first 81 nodes in 5000 nodes)

We deploy 1% infectious nodes in $A_1B_1$ of PM (See Fig. 4.1). We also assume all nodes are vulnerable and no nodes are patched. Based on the different propagation probabilities, the entire IP space is divided into three ranges:

- $R_1$: $A_1B_1$
- $R_2$: $A_1B_2 \rightarrow A_1B_g$
- $R_3$: $A_2 \rightarrow A_g$

***Result*:**

As shown in Fig. 4.14, the propagation probabilities are initially almost the same for both $\beta_1$ and $\beta_2$ via 28 intermediate nodes. Later, however, the propagation probabilities decrease gradually. This matches the real spreading tendency in [2] quite well.

We also observe the effect of different impact factors step by step. In Fig. 4.15, after 60 hops the propagation probability approaches zero, which indicates the worm theoretically propagates in a limited range of vulnerable nodes. This is in accordance with the real case analysis by [2] as the propagation time delay largely increases because of network congestion,

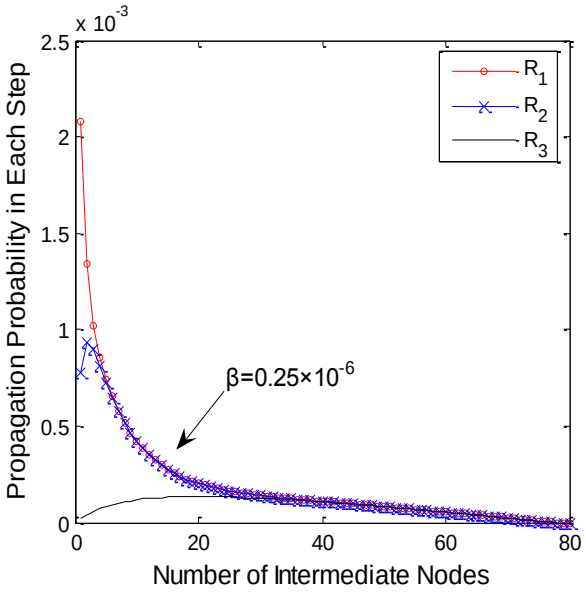



Figure 4.15: Effect of impact factor $\beta$ on propagation probability in each time unit (the first 81 nodes in 5000 nodes)

and thus an infectious node cannot spread worms to the target. Therefore, the propagation probability is close to zero

***Discussion*:**

The impact factor $\beta$ is to reflect the mutual impact between the propagation probability and the time delay. When $\beta$ increases, from Fig. 4.14, the time delay has a greater impact on the spreading of the worm, which results in a decrease of the propagation probability. If the value of $\beta$ increases continuously, the time delay will increase and the worms will not be able to propagate to the target, which reflects real scenarios. Moreover, according to Fig. 4.2(b) and Fig. 4.15, we find that an increase of $\beta$ leads to the propagation probability decreasing gradually and tending towards zero. This also indicates that an increase in time delay results in a small propagation probability of the worm's propagation. However, in the real world, each well-known worm has its own feature for propagation. How to formulate the value of $\beta$ to accurately reflect the characteristic of propagation is an issue of modeling a worm's propagation that we will address in the future.

## 4.5 Discussion of the Overestimation in the Macroscopic Model

Scanning worms infect targets by scanning the entire network and probing for vulnerable machines. Many researchers have studied and modeled the propagation of various worms using a variety of approaches and a number of different modeling techniques that address particular problems being examined. In this chapter, we generalize previous works, such as [2, 7, 26, 56, 60] as macroscopic models and propose our microscopic modeling method. Macroscopic models rely on differential equations to predict worm behavior and can effectively identify the spreading tendency of worms and their infection scale along with the elapsed time. Our proposed microscopic model adopts matrix computation and focuses on presenting the propagation procedure of worms. In the remainder of this section, we will analyze the overestimation in traditional macroscopic modeling methods which can be avoided in the microscopic point of view, and thus, is a key reason why we chose the microscopic modeling approach.

Macroscopic methods model the propagation of worms through observing the current number of infected hosts and identifying the number of possible hosts for immediate and subsequent infection. These methods construct differential equations as a function of time $t$ to calculate the number of possible hosts that can be infected in each time tick. The propagation analysis of macroscopic models starts from a group of infected nodes and this group is updated by conducting the propagation from infected nodes to uninfected vulnerable nodes, which are used again as initial infected nodes for propagation. This process continues as time elapses, ad infinitum. In our proposed microscopic model, we simulate the propagation of worms by constructing the spreading path from the initially infectious nodes to the targets via some intermediate nodes. According to the microscopic modeling and analysis of the propagation procedure, we have found an important source of inaccuracies in macroscopic

modeling caused by propagation cycles (Section 3.3.4 in Chapter 3). These propagation cycles lead to overestimation in the macroscopic analysis of worms propagation. This is one of the reasons why we believe our microscopic model performs better than previous models.

In this chapter, we focus on scanning worms that primarily belong to the non-reinfection class of worms. These types of worms, which include Code Red, can only be infected once in a worm outbreak. According to previous analyzes, this leads to overestimation due to propagation cycles among the intermediate nodes. In this section, we use a simple scenario to analyze the errors.

***Preparation*:**

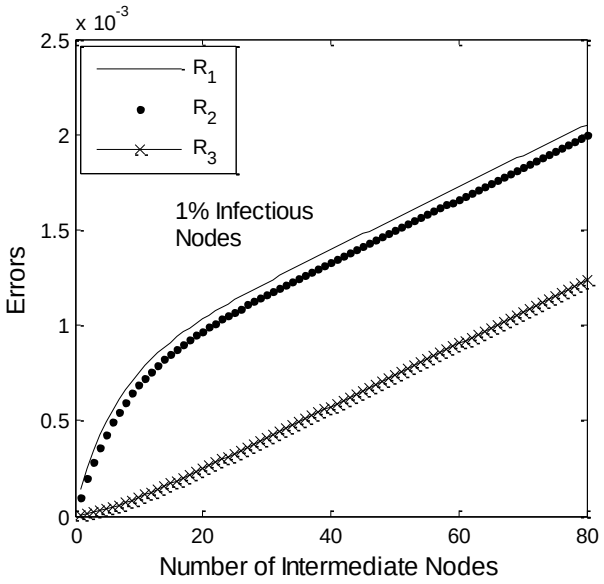
We deploy 1% of infectious nodes in $A_1B_1$ of PM (See Fig. 4.1). We also set all nodes as vulnerable and set no patched nodes. Based on the different propagation probabilities, the entire IP space is divided into three ranges:

- $R_1$: $A_1B_1$
- $R_2$: $A_1B_2 \rightarrow A_1B_g$
- $R_3$: $A_2 \rightarrow A_g$

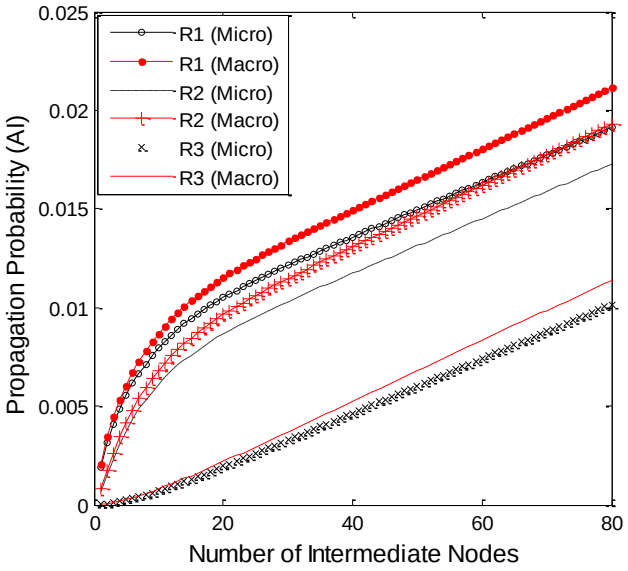
***Result*:**

As shown in Fig. 4.16(a), errors occur in different IP ranges during the propagation procedure: Errors ($R_1$) > Errors ($R_2$) ≫ Errors ($R_3$). Within the first 20 iterations, $R_1$ increases rapidly, while $R_3$ remains stable. In our microscopic model we remove the errors. Fig. 4.16(b) shows the propagation probability AI in different IP ranges before and after the removal of errors when the worm's propagation is via some intermediate nodes. From the curves, we find the noticeable differences.

***Analysis*:**




(a)




(b)

Figure 4.16: Errors analysis (the first 81 nodes in 5000 nodes)

Fig. 4.16 demonstrates that non trivial differences exist between macroscopic and microscopic models. This difference is accounted for by errors introduced by propagation cycles in the macroscopic model. According to (Formula (3.18) in Chapter 3), errors are mainly composed of two parts: the propagation probability from node *s* to node *i* when iterated (*k* - *x*) times and the propagation probability from node *i* to node *i* when iterated *x* times. In Fig. 4.16(a), the errors have curves analogous to AI, but which are two magnitudes

smaller ($10^{-3}$ compared with $10^{-1}$). In the experiments, similar results also exist in other scenarios. In Fig. 4.16(b), we show that when a worm starts to propagate, more intermediate nodes are involved in the worm's propagation. This results in the continuous and increasing formation of propagation. Thus, the errors increase rapidly especially when the worm spreads via the first 20 intermediate nodes. Then, when more vulnerable nodes in the network are infected, the growth of propagation cycles tends to stabilize. Consequently, the errors increase slowly. After eliminating the errors, we find a clear difference in each IP region. Through the inspection of these errors, however, we can eliminate this negative effect using (Formula (3.19) in Chapter 3).

Moreover, in Fig. 4.16(b), we can see noticeable differences between the macroscopic model and the microscopic model. Although the magnitude of errors is small ($10^{-3}$), we cannot regard them trivially when more initial infectious nodes or a larger network is involved. Especially for security companies, the errors can possibly mislead analysis on predicting the infected scale of the network and even cause a significant economic loss.

## 4.6 Discussion and Open Issues

Several limitations and open issues are worth discussing. First, the microcosmic model is not a complete substitute for the traditional macroscopic model of worm propagation. In order to provide an insight into the change of propagation probability between nodes, the propagation source *S* in our model has been constructed according to different initial scenarios. Thus, *S* is static. However, in the traditional macroscopic models [2, 6], the infectious state is a function of time *t* allowing that the traditional models dynamically reflect the changes during propagation. These two approaches model worm propagation from different perspectives and both are useful in worm analysis.

Second, our model employs an *n* by *n* square complex matrix to describe a network, which makes two arbitrary nodes adjacent. Thus, this representation is suitable for worms that scan the entire network and spread themselves to the target without regard to topological constraints. In the real world, some worms, such as email worms, are dependent on the topology of a network in infecting targets. Our model cannot directly simulate these worms, however, if we assume the value of propagation probability in our proposed model as being either one or zero to indicate the existence or non-existence of a directed link between the nodes, then we can extend our model to simulate the topology-dependent worm propagation.

Third, many corporations prioritize the patching of various vulnerabilities on the basis of their own vulnerability ranking system. For example, vulnerabilities in firewalls should be patched as soon as possible because firewalls directly face the internet. Our microscopic model cannot describe this type of context dependent information. We believe this issue requires additional knowledge and is out of scope of this investigation.

Fourth, in this chapter, we have not thoroughly investigated the impact factor $\beta$ and the effect of errors. In fact, subtle changes in these may result in perceptible variances. This particularly happens in large scale worm propagation. However, like the undiscovered parameter $\alpha$ in [6], we do not know the exact value of $\beta$ for real world worm propagation. More research and discussion will address these two factors in our future work.

Finally, in the experiments, we found that the overhead for the simulation is high. Given that industry has existing infrastructure in clouds and cluster environments, accuracy in the worm propagation model is the key component to be addressed compared to the issue of time cost. In future work, we will employ more practical analysis of parallel algorithms to implement our model.

## 4.7 Summary

In this chapter, we used Code Red II as an example to evaluate the vulnerable distribution and patch strategy vector in the proposed microcosmic model (in Chapter 3) and presented a series of recommendations and advice for immunization defense. Firstly, if the propagation sources can be predicted, the best strategy is to patch nodes that have IP addresses in the same net block. Otherwise, the IP ranges with a high density of vulnerable nodes are essential areas for patching. Secondly, for high risk vulnerabilities, it is critical that networks reduce the number of vulnerable nodes to below a certain threshold, e.g., 80% in this analysis. Thirdly, increased disclosure of specific vulnerabilities could possibly be delayed until the patching rate reaches a certain threshold, e.g., at least 20% in this analysis. Furthermore, we discuss the effect of the impact factor that reflects the impact of propagation time delay on the propagation probability and the overestimation in macroscopic models caused by propagation cycles.

The proposed theoretical design and experiments are based on typical scanning worms. However, there are also topology-based worms that are actively used throughout the internet. Thus, our future work will mainly focus on modeling the propagation of topology-based worms.

# Chapter 5

# Modeling of the Propagation and Defense Study of Topology-based Worms

Topology-based worms, such as email worms, pose a critical security threat to the Internet and thus, large amounts of money and labor are spent on controlling and reducing the impact of their outbreak each year. These worms rely on searching for local information to uncover the local communication topology and find new victims. Through an accurate and realistic modeling of the propagation process, we may devise effective strategies for defense and reduce such expenses. In order to access the propagation accurately and address effective schemes to deal with the problems of where and how many nodes we need to patch, we particularly focus on the spreading process of topology-based worms between each pair of nodes. We implement a series of experiments to evaluate the effects of each major component in the proposed model for topology-based worms. From the results, the network administrators can make decisions on how to immunize the highly-connected node for preventing the propagation of topology-based worms.

## 5.1 Introduction

Topology-based worms, such as email worms and social network worms, rely on email address books or friends lists contained in the victim hosts' hard drive to locate new targets and further require human interaction to spread. Typical examples are worms such as the "Here you are" email worm [58] and Koobface [27], which emerged on Facebook in recent years. Spreading can take place rapidly and leads to potential network damages and service disruption. According to the official Internet threats report of Symantec Corporation [59], topology-based worms and resembling attacks accounted for 1/4 of the total threats in 2009 and nearly 1/5 of the total threats in 2010.

Different from the propagation of scanning worms, topology-based worms pose a significant threat to the network where topologies play an important role for worms propagation. Firstly, worms search for local information to find new targets by trying to discover the local communication topology. This allows a topology-based worm to be far more ***efficient*** than a scanning worm as it does not make a large number of wild guesses for every successful infection. Instead, it successfully infects on most attempts. This makes topology-based worms less vulnerable to containment defenses based on looking for missed connections or too many connections. Secondly, topology-based worms can potentially be very fast. They rely on the information contained in the victim machine to locate new targets. This self-broadcast mechanism allows for the worm's ***rapid*** reproduction and spread. Thirdly, due to social engineering techniques, most internet users can fail to recognize the malicious code, resulting in a ***wide*** range of infection. Therefore, in order to take an effective countermeasure to prevent the propagation of topology-based worms as much as possible, we must understand the propagation mechanism.

The goal of this work is to develop a modeling framework that can characterize the spread of topology-based worms and provide a series of effective patching strategies which will benefit IT industries and security best practice. To this end, we first construct the propagation mechanism of topology-based worms by concentrating on the propagation probability and model the propagation procedure through *k*-hops. With the help of the model, we then evaluate the mutual effect of initially infectious states and patch strategies. We take advantage of the propagation probability between each pair of nodes to explore the propagation procedure of worms and estimate both infection scale and defense effectiveness. Through model analysis, we derive a better understanding of dynamic infection procedures in each step rather than recapitulative analysis on the propagation tendency [6, 23-25, 60]. Specifically, we aim to understand: 1) the propagation probability between each pair of nodes; and 2) how one node infects another node through a group of intermediate nodes.

The rest of this chapter is organized as follows. We provide related work in Section 5.2. In Section 5.3, we propose the propagation model for topology-based worms and introduce each component of the model. Then, we conduct an analysis and deduce the result for obtaining an optimized patch strategy in Section 5.4. Section 5.5 discusses the formation of propagation errors and examines the impact of eliminating errors on the propagation procedure. Finally, we conclude the chapter in Section 5.6.

## 5.2 Related Work

In the area of network security, several approaches have been proposed to model and simulate the spreading of worms in the network.

The classical deterministic epidemic models [13, 24] are Susceptible-Infectious (*SI*) models, in the sense that all hosts can have only one of two states: susceptible or infectious.

Staniford *et al.* [7] presented a random constant spread model (RCS) for the Code-Red I v2 worm. Essentially, it is the above classical simple epidemic model allowing for the infection rate to be constant, and without considering patching cases. The classical general epidemic models [26] improve the classical simple epidemic models by considering the removal of infectious hosts due to patching. Zou *et al.* [2] proposed a two-factor model on the basis of the classical simple epidemic model. This model introduced human countermeasures in patching, the removal of hosts from both the infectious and susceptible population, and considered the infectious rate as a variable but not a constant. Additionally, models from Z. Chen *et al.* [10] and Y. Wang *et al.* [61] took into account the time taken to cause an infection from spreading the virus from one infected host to other hosts. However, all of the above models rely on the homogeneous mixing assumption that an infected host can infect any other susceptible hosts with equal possibility. Thus, they are no longer appropriate to model the propagation of topology-based worms since these models overestimate the worm's propagation speed, especially at the beginning stage when a small number of nodes are infected and clustered with each other [6].

K.R. Rohloff *et al.* [8] presented a stochastic density-dependent Markov jump process propagation model for RCS (Random constant Scanning) worms, drawn from the field of epidemiology. Sellke *et al.* [9] built up a stochastic branching process model to characterize the propagation of worms using a random scanning approach. It developed an automatic worm containment tactic for preventing the worm propagation beyond its early states. Nevertheless, these two models are based on a linear structure or a one-to-many hierarchy and thus, they are not applicable to topology-based worms.

A topology-based model describes the worms that rely on the topology for spreading themselves. Fan and Xiang [19] employed a logic matrix approach to model the spreading of peer-to-peer worms between each pair of all peers. They discovered the relation between out-

degree, vulnerability and coverage rate in power law and simple random graph topologies respectively. However, they did not consider the propagation probability and infected probability of each node, which had a significant impact on the infection procedure. Zou *et al*. [6] considered these two probabilities and compared internet email worm propagation on power law, small world and random graph topologies. In the proposed model, the probability of each user opening a worm attachment can be treated as an infected probability and the distribution of email checking times can represent the propagation probability. However, this model still describes the email worm propagation tendency instead of modeling the dynamic spreading procedure between each pair of nodes.

We propose a probability matrix that models topology-based worm propagation and analysis the spreading procedure of worms. Using this matrix in the propagation of worms forms the major difference between this work and existing work. In our model, we focus on investigating the procedure of worms' spreading and providing effective patching strategies for preventing topology-based worms from propagating in the network.

## 5.3 Propagation Model

In this chapter, in order to describe how topology-worms propagate in the network, we choose a typical topology-based worm on the basis of email worms which infect their logical neighbors through sending malicious email attachments.

### 5.3.1 Propagation Matrix (P)

Instead of the complex matrix in Chapter 3, we propose employing an $n$ by $n$ square matrix $P$ with elements $p_{ij}$ to describe a network consisting of $n$ nodes. We consider that two nodes in the network are connected, thereby making node $i$ and $j$ immediate neighbors. In this

matrix, each element $p_{ij}$ represents a propagation probability of the worm spreading from node $i$ to node $j$ under the condition that node $i$ is infected. We call this matrix the propagation probability matrix ($P$) of network, as shown in (5.1).

$$P = \begin{bmatrix} p_{11} & \cdots & \cdots \\ \cdots & p_{ij} & \cdots \\ \cdots & \cdots & p_{nn} \end{bmatrix}_{n \times n} \qquad (5.1)$$

$$p_{ij} = p(N_j | N_i) \quad p_{ij} = 0 \ (i = j), \ \sum_{j=1}^{n} p_{ij} \in [0,1]$$

where $N_i$ denotes the node $i$, $N_j$ denotes the node $j$ in the network. Each row of the $P$ represents the propagation probability from one infected node to all other nodes. Each column of the $P$ represents the propagation probability from infected nodes to a target node. We assume one node cannot propagate the worm to itself, so the probability of self-propagation is zero.

### 5.3.2 Propagation Probability

In real-world conditions, worms can be spread between nodes from node $i$ to node $j$ via one or more intermediate nodes. In existing worms it is observed that an infectious node can propagate worms and a vulnerable node can also be infected and become a new infectious node for future propagation with a certain probability.

We assume that a worm's propagation from node $i$ ($N_i$) to node $j$ ($N_j$) is via and only via $k$ intermediate nodes in a network consisting of $n$ nodes. It is denoted by $p_{ij}^{(k)}$ and defined in (5.2):

$$p_{ij}^{(k)} = \sum_{m=1}^{m=n, m \neq i} p_{im}^{(k-1)} p_{mj}, k \in [1, n-2], \quad i = 1,...,n, \qquad j = 1,...,n \qquad (5.2)$$

Since $N_i$ self-propagation via $k$ nodes is meaningless in the real world, we let the value of this propagation probability be zero; namely $p_{ij}^{(k)}$ =0 when $i$=$j$. We introduce a function $\gamma$ to conduct the iterated procedure. It is defined in (5.3):

$$\gamma^0(P) = P, \quad \gamma^k(P) = \underbrace{P \times P \times \cdots \times P}_{k+1} \tag{5.3}$$

Operation $\times$ is the traditional matrix multiplication. Subsequently, the $P$ can be represented by the following equation when the worm's propagation is via and only via $k$ intermediate nodes, as shown in (5.4):

$$P^{(k)} = \begin{bmatrix} p_{11}^{(k)} & \cdots & \cdots \\ \cdots & p_{ij}^{(k)} & \cdots \\ \cdots & \cdots & p_{nn}^{(k)} \end{bmatrix}_{n \times n} = \gamma^k(P) \tag{5.4}$$

### 5.3.3 Propagation Time

In real scenarios, topology-based worms attack victims in the network via neighbor lists. For example, email worms search all email addresses found on the compromised user's computer to spread themselves. Social network worms look for the friends' list from the victim's account and use this list as targets. In this study, we mainly focus on the propagation procedure of worms and thus, we assume all events (worm infection, user checking email, user clicking website, etc.) happen right at each discrete time tick. Once a host is infected, it immediately sends out malicious messages to its neighbors at time tick $t$ and the messages could be read by its recipients as soon as the next time tick, $t$+1. Therefore, the propagation time of topology-based worms in the proposed model is equivalent to be presented by the number of intermediate users in the propagation path from initially infectious users to the current infected user.

For email worms, if user *i* checks email at time *t*, the user checks all new email received after his or her last email checking. When a worm email is opened, user *i* is infected and the worm will send a worm email to all neighbors of the user. These worm emails are read at the next time tick. Thus, the time of a current infected user *j* being infected by user *i* is represented by the hops from user *i* to user *j*.

### 5.3.4 Propagation Source Vector (*S*)

In a network, a propagation source is one of the significant factors for worm propagation, which represents whether the state of the node has been infected or not. An initial propagation source vector (*S*) is defined as shown in (5.5). An infectious node that can propagate worms is represented with a probability of one. Zero means that a node is healthy and does not have the ability to propagate the worm.

$$S = \left[s_1, s_2, \ldots, s_i, \ldots, s_n\right]^T, \ \ s_i = 0 \ \ or \ \ 1, \quad i = 1 \ldots . n \tag{5.5}$$

The iterated procedure can be represented as function $\gamma_s$ in (5.6):

$$\begin{aligned} &\gamma_s^{\,0}(P) = S \,\&_L\, P \\ &\gamma_s^{\,k}(P) = \gamma_s^{\,k-1}(P) \times P = (S \,\&_L\, P) \times \underbrace{P \times \cdots \times P}_{k} \qquad (k \geq 1) \end{aligned} \tag{5.6}$$

We define $\&_L$ the same as in formula (3.7) in Chapter 3. During the propagation process, each intermediate node can be infected and become infectious. We introduce an infected probability vector *I*, as shown in (5.7):

$$I_s^{(k)} = \left[S^T \times \gamma_s^{\,k}(P)\right]^T \qquad (k \geq 0) \tag{5.7}$$

$I_s^{(k)}$ reflects the infected possibility of each node after a worm's propagation via *k* intermediate nodes under a certain deployment of *S*. The *P* can be represented by the

following equation when a worm's propagation is via and only via *k* intermediate nodes, as in (5.8):

$$P_s^{(0)} = \gamma_s{}^0(P), P_s^{(k)} = \gamma_s^k(P) = P_s^{(k-1)} \times P \qquad (k \geq 1) \tag{5.8}$$

### 5.3.5 Patch Strategy Vector (*Q*)

A patch strategy is another important factor for the propagation of topology-based worms, which provides an approach to cure infected nodes. An infected node can be cured so it is unable to spread worms to other nodes. Therefore, we need to remove these nodes from the propagation process at that time. We define a patch vector *Q* in (5.9). For each element in *Q*, the value of one represents that a node has been patched and becomes a healthy node. A value of zero indicates that a node is still vulnerable.

$$Q = \left[q_1, q_2, \ldots, q_i, \ldots, q_n\right]^T, \; q_i = 0 \; or \; 1, \quad i = 1 \ldots . n \tag{5.9}$$

Once the nodes have been patched, they will be immune to the worms and lose their infectious ability. Thus, we should exclude these patched nodes in the matrix for the successive iteration. The iterated procedure can be represented as function $\gamma_{sq}$ shown in (5.10):

$$\begin{aligned} &\gamma_{sq}{}^0(P) = S \,\&_L\, P \,\&_R\, Q^T \\ &\gamma_{sq}{}^k(P) = \gamma_{sq}{}^{k-1}(P) \times (P \,\&_R\, Q^T) \quad (k \geq 1) \end{aligned} \tag{5.10}$$

We define $\&_R$ the same as formula (3.13) in Chapter 3. After considering the *Q*, the *P* and infected probability vector *I* can be represented by the following equations respectively when the worm propagates via and only via *k* intermediate nodes, as shown in (5.11):

$$P_{sq}{}^{(k)} = \gamma_{sq}{}^k(P), \quad I_{sq}^{(k)} = \left[S^T \times \gamma_{sq}{}^k(P)\right]^T (k \geq 0) \tag{5.11}$$

### 5.3.6 Accumulative Infected State (*AI*)

We introduce an infected probability vector *I* for evaluating the infected capability of each node in the network, as shown in (5.12):

$$I = [i_1, i_2, \ldots, i_x, \ldots, i_N]^T, \quad i_x = 1 - \prod_{k=0}^{K} [1 - s_x^{(k)}] \tag{5.12}$$

where *K* means the maximum number of intermediate nodes when no nodes can become infectious.

In consideration of more than one path for the propagating worm, we adopt an accumulative infected state (*AI)* to represent the sum of probabilities for the worm propagation via at most *k* intermediate nodes. It is defined in (5.13).

$$AI^k = \frac{\sum_{x=1}^{n^k} i_x}{n^k} \tag{5.13}$$

## 5.4 Model Analysis

### 5.4.1 The Experimental Environment

Our implementation is in Visual C++ 2008 SP1 and Matlab 7. The random numbers in our experiments are produced by the C++ TR1 library extensions. In order to show these results clearly, we choose reasonable network sizes (5000 nodes) and examine these under different scenarios.

In our experiments, we use a typical topology-based worm on the basis of email worms to investigate the propagation procedure. The topology of an email network has been studied by

many researchers [6, 62] because it plays a critical role in determining the propagation dynamics of an email worm. According to the analysis in [6, 33, 62], the topology mainly has the characters: 1) the topology can be thought of as a "semi-directed network", a graph in which some edges are directed and others are undirected; 2) users who have large groups of friends tend to appear in the contact lists of many others; 3) the weight of each edge denotes the probability of a user being infected by one of their friends. This probability is strongly affected by human factor. Therefore, we let the topology of the network in the experiments follow Power Law Distribution, namely nodes with the higher value of topology out-degree are the minority, most nodes having a lower value of topology out-degree. We assume checking probability of email follows the Gaussian Distribution $T \sim N(0.5, 0.2^2)$.

#### *5.4.1.1 Power-law Network Generator*

The Power Law Topology can be characterized by the following equations, as shown in (5.14). We assume *P(x)* follows power law distribution.

$$\begin{aligned} &P(x) = Cx^n, x \in [x_0, x_1] \quad n < 0 \\ &\int_{x_0}^{x_1} P(x)dx = C\frac{\left[x^{n+1}\right]_{x_0}^{x_1}}{n+1} = 1 \\ &C = \frac{n+1}{x_1^{n+1} - x_0^{n+1}} \end{aligned} \tag{5.14}$$

where $x_0$ and $x_1$ represent the minimum topology out-degree and maximum topology out-degree respectively, $n$ is the power law exponent, and $C$ is a constant. In order to make sure the topology of networks follows power law distribution, we deduct the out-degree of each node by the following equations, as shown in (5.15) and (5.16). Firstly, we assume $y$ is a uniformly distributed variant on [0, 1]:

$$y \equiv D(x) = \int_{x_0}^{x} P(x')dx' = C\int_{x_0}^{x} x'^n\, dx' = \frac{C}{n+1}\left(x^{n+1} - x_0^{n+1}\right) \tag{5.15}$$

Then, according to the (21), we can derive the out-degree *x* of each node that follows power law distribution strictly from the following equation:

$$x = \left( \frac{n+1}{C} \cdot y + x_0^{n+1} \right)^{1/n+1} = \left[ \left( x_1^{n+1} - x_0^{n+1} \right) \cdot y + x_0^{n+1} \right]^{1/n+1} \tag{5.16}$$

Finally, once the power law exponent *n* is determined, given $x_0$ , $x_1$ and a uniformly distributed variant *y*, the out-degree of each node in the network can be worked out, which models the Internet email network followed the power law topology distribution.

#### *5.4.1.2 Effect of Power Law Exponent* **n**

The power law exponent *n* is an important parameter for a power law topology. Combining with the minimum and the maximum topology out-degree, it limits the expected value of topology out-degree [26], as shown in (5.17).

$$E(d) = \sum_{x=x_0}^{x_1} x^{1+n} / \sum_{x=x_0}^{x_1} x^n \tag{5.17}$$

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

In a real Internet email network, the true value of *n* is variable. In order to observe how the power law exponent *n* affects the power law topology, we compare the changes of *E(d)* under a different value of *n*. In a real scenario, a key user has some possibility to connect all other users. According to the different topology, the value of out-degree is contingent. In our experiment, we assume the minimum topology out-degree x0 is equal to 3, the maximum topology out-degree increases from 100 to 550 with step size 50. We believe the range of 100 to 550 is a reasonable area. The result is shown in Fig. 5.1, which reveals that a larger maximum topology out-degree requires a larger power law exponent *n*, and that a larger expected value of topology out-degree demands a smaller power law degree.

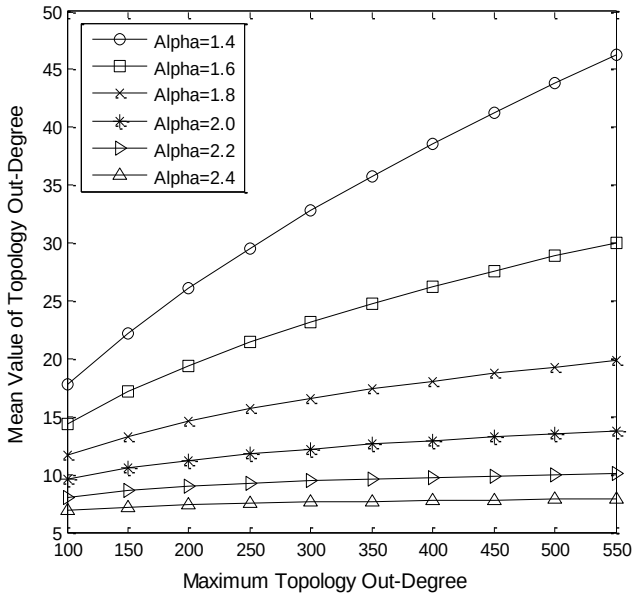



Figure 5.1: Power law exponent *n*

## 5.4.2 Effect of the Propagation Source Vector

In this subsection, we assume all nodes in the network are vulnerable and no nodes have been patched. Since an email worm's propagation depends on a different topology of the network and has a close relation with the out-degree of initially infectious sources, we arrange a group of scenarios with practical meaning in Table 5.1 to describe the worm's spreading under different origins of worms. The results are represented by the maxima and the minima of the value of infected probability: *AI*(Max), *AI*(Min).

Table 5.1 Scenarios for Analyzing Infectious Source (*S*) in Email Worms

| ***Scenario*** | ***Description*** | ***Practical Meaning*** |
|---|---|---|
| 1 | In a ***low*** expected out-degree network, the initial infectious node has the highest degree or has the lowest degree in the topology. | Analyzing the impact of initially infectious sources located in the key user or normal user on the Email worm's propagation in a ***sparsely*** connected Email Community. |
| 2 | In a ***high*** expected out-degree network, the initial infectious node has the highest degree or has the lowest degree in the topology. | Analyzing the impact of initially infectious sources located in the key user or normal user on the Email worm's propagation in a ***densely*** connected Email Community. |

Table 5.2 Scenario 1: A list of AI ($\alpha$ = 2.2)

| ***Max (OD)*** | ***Average (OD)*** | ***Infected Probability*** | | | |
|---|---|---|---|---|---|
| | | ***AI(maxOD)*** | | ***AI(minOD)*** | |
| | | ***Minima*** | ***Maxima*** | ***Minima*** | ***Maxima*** |
| 100 | 8.04 | 0.2450 | 0.9408 | 0.1138 | 0.9270 |
| 200 | 8.93 | 0.3215 | 0.9601 | 0.1201 | 0.9488 |
| 300 | 9.41 | 0.3263 | 0.9649 | 0.1287 | 0.9558 |
| 400 | 9.73 | 0.3585 | 0.9711 | 0.1575 | 0.9603 |
| 500 | 9.96 | 0.3912 | 0.9766 | 0.1585 | 0.9667 |

OD: the value of out-degree

#### *5.4.2.1 Scenario 1*

We consider two cases: in the first case we select the node with the highest out-degree as an initially infectious node, while in the second case the initially infectious node has the lowest degree. We fix the number of initially infectious nodes to be one in both cases. Both power law networks have the same nodes and a power law exponent of $\alpha$=2.2, which represents a *sparse* connection. From Table 5.2 and Fig. 5.2, *AI* declines sharply at the beginning stage in both cases, then *AI* increases continuously when more intermediate nodes become involved, and finally achieves the maxima. When the average out-degree increases, *AI* in both cases goes up. The maxima and minima of *AI* in the first case are larger than in the second case.

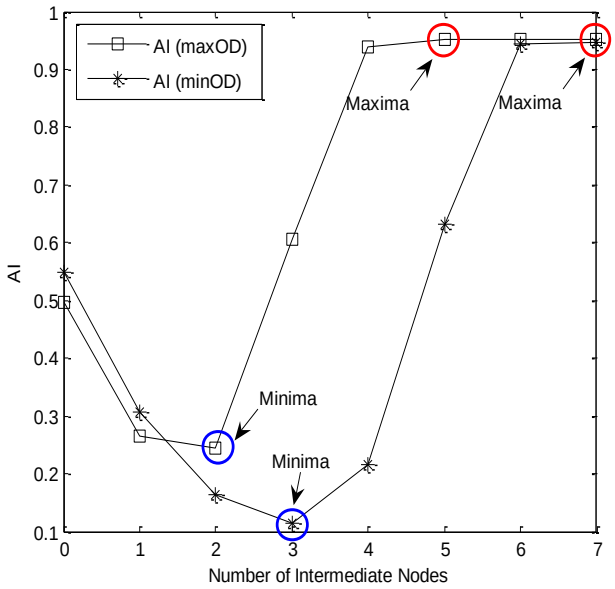



Figure 5.2: Propagation probability in scenario 1

***Analysis*:**

At the beginning the infected probability declines sharply because the infected probability of a node is smaller than when it is directly infected. However, when some intermediate nodes with high out-degree are involved, a large number of nodes are infected quickly, which results in the *AI* increasing continuously. The *AI* will increase if the maximum out-degree increases, meanwhile the number of intermediate nodes decreases meaning more nodes can possibly be infected via less intermediate nodes when the network has a high average out-degree. From Table 5.2, we also observe that when the initially infectious node has a higher out-degree, the infected probability of nodes will be larger than the node that has a lower out-degree. This shows that a worm's propagation can be effectively prevented if we immunize the infected nodes with a higher out-degree.

### *5.4.2.2 Scenario 2*

Similar to scenario1, we consider the initially infectious node has the highest out-degree (the first case), and it has the lowest degree (the second case). In both cases, the number of

Table 5.3 Scenario 2: A list of the AI ($\alpha$ = 1.6)

| ***Max (OD)*** | ***Average (OD)*** | ***Infected Probability*** | | | |
|---|---|---|---|---|---|
| | | ***AI(maxOD)*** | | ***AI(minOD)*** | |
| | | ***Minima*** | ***Maxima*** | ***Minima*** | ***Maxima*** |
| 100 | 14.33 | 0.3039 | 0.9431 | 0.1218 | 0.9289 |
| 200 | 19.42 | 0.3466 | 0.9700 | 0. 1564 | 0.9616 |
| 300 | 23.16 | 0.4139 | 0.9711 | 0.1632 | 0.9617 |
| 400 | 26.22 | 0.4720 | 0.9741 | 0.2685 | 0.9660 |
| 500 | 28.86 | 0.5366 | 0.9842 | 0.3204 | 0.9700 |

OD: the value of out-degree

initially infectious nodes is one. Both power law networks have the same nodes (5000 nodes) and a power law exponent of $\alpha$=1.6, which represents a *dense* connection. From Table 5.3, we observe that the change of *AI* is similar to Scenario 1.

***Analysis*:**

Scenario1 and 2 discuss the infected probability of nodes (*AI*) when worms propagate in different connection densities. Compared with scenario1, the AI in scenario 2 is larger. This means when worms propagate in a densely connected network, more nodes can be infected than worms spread in a sparsely connected network no matter if the initially infectious node has the highest or the lowest out-degree. Moreover, regardless of connection densities, more nodes can possibly be infected if the initially infectious node has a higher out-degree. Therefore, in a densely connected network, if the highly-connected infectious node is immunized, the worm spread will slow obviously.

### *5.4.2.3 Inspiration for Developing the Patch Strategy*

The two different cases indicate whether the node is a key user or a normal user. The two different scenarios show the connection densities of the network. In practice, a key user has a larger email list than a normal user. If the key user, especially in a highly-connected network, is compromised by malicious email worms, more normal users will be infected when they

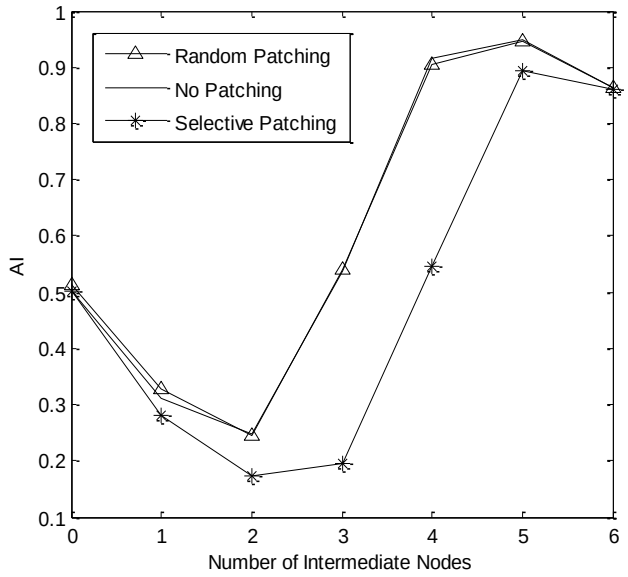



Figure 5.3: Patching strategy in email worms

open email attachments from the key user. Therefore, email worm's propagation can be effectively prevented or slowed down when key users can be immunized regularly.

### 5.4.3 Effect of the Patch Strategy Vector

In this subsection, we analyze the effect of *Q*, which is used to immunize the small subset nodes for preventing the propagation of worms effectively. The results are represented by the accumulative infected probability *AI*. In order to observe how the patch strategy impacts the worm's propagation, we consider two different immunization defense methods. In the first case, we randomly deploy 5% of nodes to patch, while in the second case we patch the 5% of highly-connected nodes in the network. In both cases, the number of initially infectious nodes is one. Both power law networks have the same nodes and a power law exponent of $\alpha$=2.2. The network has an average out-degree of 8. From Fig. 5.3, the curve of the selective patch strategy is obviously lower than the random patch strategy. However, the AI after random patching nodes is similar to no immunization.

***Analysis*:**

According to Fig. 5.3, we can observe that patching highly-connected nodes is a quite effective strategy to slow down the propagation of email worm. However, if we randomly select nodes to immunize, there is a high chance of choosing the nodes with a lower out-degree in a power law network, which results in a small group of nodes avoiding infection during worm's propagation. Thus, the effect of random patching is not obvious as email worms spread by relying on the underlying connectivity between each pair of nodes.

## 5.5 Propagation Errors

Currently, a variety of models have been proposed for modeling the propagation mechanism. A common feature of all current epidemic models [2, 5-10, 15, 23-26, 60, 63] is to estimate or predict the number of infected nodes in each time tick, and then the node will be counted as long as it is infected. An infectious node can spread worms via some intermediate nodes to itself again, which forms a propagation cycle in the spreading procedure. However, some worms, such as Melissa and Love Letter, belong to the non-reinfection class of worms. These types of worms can be infected only once. Consequently, propagation errors, and an overestimation of the scale of the infected network, cannot be avoided by previous research if a node is infected more than once.

A possible reason for the above overestimation may be rooted in the failure of considering the dynamic spreading procedure between each pair of nodes. Most current models pay attention to analyze the overall scale of the infected network, and do not investigate the concealed errors between each pair of nodes. Although [57] identified that an email network contains cycles, it only considered the topology of the email network as a tree structure. It did not discuss the errors caused by a process of self-infection through other nodes, called propagation cycles. Based on our knowledge, few models currently aim to eliminate errors.

Both macroscopic and microcosmic worm propagation models can encounter the problem of propagation cycles. Thus, we introduce an error calibration vector in the proposed microcosmic model to eliminate errors caused by propagation cycles.

In this subsection, we use the revised microcosmic propagation model to efficiently prove the existence of propagation cycles and consequentially, the propagation errors caused by them. We discuss the negative effects of the errors and propose a method to remove it by using our formulized definition. Validation against conducted simulation experiments reveals that our analysis of errors helps correctly estimate the worm's spreading.

***Preparation*:**

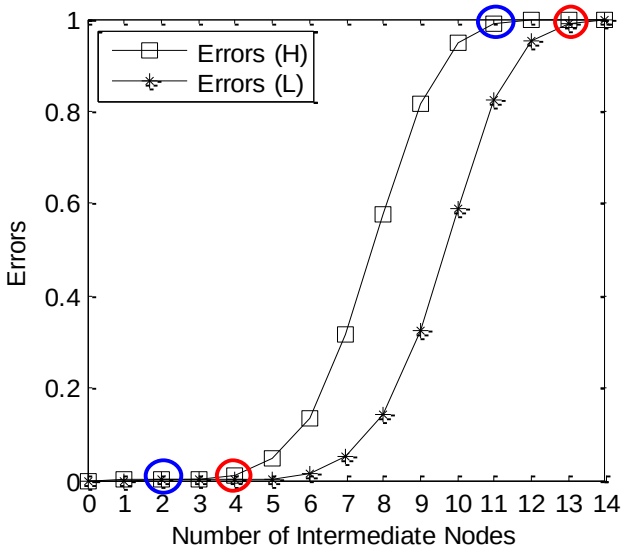
To evaluate the propagation errors caused by cycles, we introduce an error calibration vector *E* when there are *k* intermediate nodes, as shown in (5.18).

$$E^{(k)} = \left[e_1^{(k)}, e_2^{(k)}, \ldots, e_x^{(k)}, \ldots, e_N^{(k)}\right]^T \tag{5.18}$$

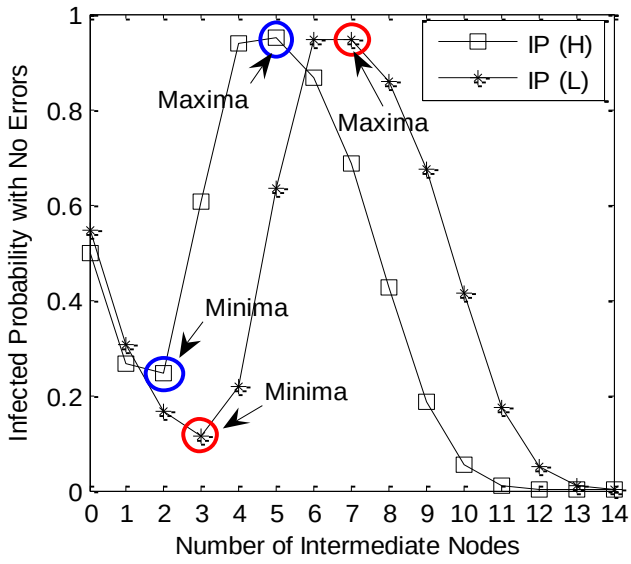
In the real word, the propagation of email worms is related to the topology of a network and the probability of opening an email. We assume the network topology follows a power law distribution and the probability of checking an email (*C*) follows a Gaussian distribution: $C \sim N(0.5, 0.2^2)$. Therefore, the propagation errors of non-reinfection email worms can be defined by (5.19)

$$e_x^{(k)} = 1 - \prod_{j=1}^{k-2}\left(1 - c_{sx}^{(k-j-1)} c_{xx}^{(j)}\right), \quad x = 1....n, k \geq 2 \tag{5.19}$$

where *k* is the number of intermediate nodes. $c_{sx}^{(k-j-1)}$ is the propagation probability from node *s* to node *x* via (*k-j-1*) intermediate nodes. $c_{xx}^{(j)}$ is the propagation probability from node *x* to node *x* via *j* intermediate nodes.




(a)




(b)

Figure 5.4: Errors analysis of non-reinfection email worms

We conducted a simulation with a power law exponent of $\alpha$=2.2. The highest out-degree of this network was 100 and the lowest out-degree was 3. We arranged the initially infectious node to have the highest and the lowest out-degree in alternate scenarios.

***Analysis*:**

In Fig. 5.4(a), if the initially infectious node has the highest out-degree, errors occur when the worm's propagation is via two intermediate nodes and reaches 100% when the number of intermediate nodes is 11. If the initially infectious node has the lowest out-degree, the errors

occur when the worm's propagation is via four intermediate nodes and reaches 100% when the number of intermediate nodes is 13. The propagation errors continuously increase in relation to the increased possibility of cycles forming when more intermediate nodes are involved. Since a node can be infected only once, when all nodes in the network have been infected, the 100% probability of nodes infecting other nodes is caused by errors only.

Fig. 5.4(b) shows the infected probability of nodes after the removal of errors when the worm's propagation is via some intermediate nodes. Because the infected probability of a node is smaller than when it is infected directly, at the beginning the infected probability declines sharply. However, when some intermediate nodes with high out-degree are involved, a large number of nodes are infected quickly, which results in the infected probability and the errors of nodes continuously increasing. When most nodes have been infected, the infected probability of nodes tends to be zero after eliminating the errors.

From Fig. 5.4, when the scale of the infected network increases, the errors will significantly mislead the analysis. Through the inspection of these errors, however, we can eliminate this negative effect. The errors can be subtracted by formula (5.19) for email worms. After the scale of the infected network reaches a peak, the infected probability of nodes declines sharply which limits the infected scale extending infinitely.

## 5.6 Summary

This chapter presented a novel process modeling the propagation of topology-based worms by concentrating on the propagation probability. In order to understand the propagation procedure, we used a typical topology-based worm, an email worm, as an example to investigate how a worm spreads from one node to another node through a group of intermediate nodes. According to the email user's behavior, such as checking email

probability, we examined the propagation source and patch strategy vector for investigating a more effective patch strategy for preventing worms from spreading. We found that for a power law network, a more effective patch strategy against an email worm's propagation is to immunize the most-connected nodes.

We also analyzed the formation of propagation errors and examined the impact of eliminating errors on the propagation procedure of email worms. Through the use of simulations, we have shown that errors increase as more propagation cycles are formed and we quantified the errors under different propagation scenarios. This work is helpful in the accurate analysis of worm spreading.

# Chapter 6

# Modeling Propagation Dynamics of Email Worms

As one of the major forms of worms, email worms pose a critical security threat to the Internet. This is because an email worm sends itself to the email addresses found on an infected computer and email recipients often trust the emails received, especially from their friends. Almost everyone uses an email service and thus, the propagation of email worms can be incredibly fast and cause significant damage. Modern email worms are more sophisticate and intelligent. For example, reinfection email worms will send malicious copies every time the user opens the worm email and self-start reinfection email worms can be triggered by specific events and the system restart process. The proposed microcosmic worm propagation model in Chapter 3 may not simulate the propagation procedure accurately. In this chapter, we present an analytical model on the propagation dynamics of email worms. Our model distinguishes itself from previous models because: 1) we extensively investigate classes of real-world worms based on their infection strategies, including non-reinfection, reinfection, and modern self-start reinfection categories; 2) we investigate the details of the propagation mechanisms by examining the individual steps and state transitions. Our model can provide

an accurate representation of the propagation of worms with different checking time of mailboxes from users; 3) our model reflects the repetitious email sending process in reinfection and self-start reinfection worms. To highlight the advantage of our analytical model, we implement a series of experiments. The results show that our modeling is accurate and can aid a better and more realistic understanding of the propagation of worms. This has benefits for devising new tactics against email worms.

## 6.1 Introduction

For a number of years, the propagation of email worms has followed the same modus operandi; a worm email is sent to victims which looks legitimate. The email appears as though it was sent by somebody the recipient trusts and the subject matter will often be related to the recipient's area of business. Once the victim is fooled into either clicking a malicious link or opening a malicious attachment, the victim's PC will be infected and start to search for local information, such as an email address book, in order to discover the communication topology of the network and infect new targets. From Melissa in 1999, Love Letter in 2000, Mydoom in 2004 and W32.Imsolk in 2010, we have witnessed the prevalence of email worms and as a consequence, the damage to the Internet. According to the Symantec Internet Security Threat Report [59], in the last two years email worms or resembling attacks accounted for 1/4 of the total threats in 2009 and nearly 1/5 of the total threats in 2010.

Although worm propagation through email is an old technique, it is still worthy of further study. Firstly, email worms would not have been successful without convincing users that the links and attachments they received in an email were from a trusted source. Unfortunately, however, most of the email recipients have little security awareness since they always trust emails, especially from their friends. Currently, almost everyone using a computer uses an

email service, which means the potential damage from email worms is likely to continue in the future. Thus, it is of significant importance to investigate email worms and how they propagate. Secondly, email worms collect information on the communication of victims. This mechanism is similar to certain types of worms like Koobface [27] spreading on social networks or Commwarrior [85] propagating through a multimedia messaging service or through Bluetooth of mobile devices. The research on the propagation of email worms can help us characterize the propagation dynamics of those isomorphic worms.

In this model, real-world email worms are classified into the following categories based on their infection strategies:

- *Non-Reinfection*: Non-reinfection means each infected user sends out worm copies only once, after which the user will not send any further worm emails, even if he opens a worm attachment again. Non-reinfection worms mainly appear in the early worm cases, such as Melissa [70] and Love Letter [71].
- *Reinfection*: Reinfection means that an email user will send out worm email copies whenever he opens an email worm attachment. Reinfection greatly accelerates the spreading speed.
- *Self-start Reinfection*: Evolving from reinfection, modern email worms modify the registry entries and can be triggered whenever the computer is restarted or certain files are opened, such as opening an image file like Mydoom [72] and W32.Imsolk [58].

In order to understand and possibly address defense strategies against email worms, it is important to analyze the propagation of worms. Previous work has adopted the classical simple epidemic model [34, 53-55] and the spatial-temporal model [32]. Recently, in order to focus on realistic scenarios of email worm propagation, researchers [6] have relied on

simulation modeling rather than on mathematical analysis. The difficulty of mathematical modeling lies in two aspects. Firstly, each user has their own habits of checking emails. It is really hard to characterize the propagation dynamics with different mailbox checking time between email users in a large scale network. Secondly, modern email worms belong to reinfection or self-start reinfection worms. This means it is difficult to model the repetitious email sending process.

There are only a few email worms that attack client-side vulnerabilities in email agents and can infect computers by simply being read by users (with no attachments) [6]. In order to understand how worms propagate by email, we focus exclusively on those that propagate solely through email attachments. To facilitate an understanding of the following, if not otherwise stated, a user reading an email means opening email attachments. The motivation and contributions of our research are summarized as follows.

- We derive an accurate propagation model of email worms by observing the spreading procedure from an analytical point of view. We examine the individual spreading steps and every state transition on each node in the network so that our analytical model can reflect the propagation dynamics with the different mailbox checking habits of users (see Section 6.3).

- Zou *et al*. have mentioned a noticeable overestimation [6] in the early topological epidemic models [34, 53-55] and presented a comprehensive simulation analysis on the propagation of email worms, which has been referred to in many papers since 2007 without its accuracy being questioned. However, as we show in Section 6.5.2, their simulation model still poorly estimates the spreading speed of email worms due to their assumption regarding repetitive infectious behavior. We propose the concept of *virtual* users to represent the process of sending repetitive emails so that our analytical model can accurately reflect the propagation of reinfection worms.

- We carry out extensive studies on realistic email worms. Our contributions are made as follows:
  - As the authors in [74] stated, neither reinfection nor non-reinfection is very realistic. Indeed, according to various security reports like [59], most modern email worms that are exposed belong to the more sophisticated self-start reinfection category and use every opportunity to spread. Different from previous works, our model analyzes these types of email worms (see Section 6.6).
  - We gain insight into the trust levels among email users. In Section 6.3.1, we use a propagation matrix to present the pair-wise information between email users. Each email user has different trust levels among their friends, as opposed to a constant [6, 34, 53-55].
  - We prove the exponential increase in the number of received reinfection worm emails without considering user awareness (see Section 6.5.2). Actually, real-world email recipients may become watchful after receiving a number of emails that excessively exceeds the number they would normally receive.

The remainder of this chapter is structured as follows. In Section 6.2, we introduce related work. A basic analytical model is presented in Section 6.3. In section 6.4, 6.5 and 6.6, we model the propagation of non-reinfection, reinfection and self-start reinfection email worms respectively. We conclude this chapter in Section 6.7 with a brief summary and an outline of future work.

## 6.2 Related Work

Early research on email worm modeling mainly refers to academic thought on epidemic propagation [34, 53-55]. Distinguished by whether infected users can become susceptible again after recovery, these models can be classified into Susceptible-Infectious-Susceptible (*SIS*) models [18, 53, 74-77] and Susceptible-Infectious-Recovered (*SIR*) models [34, 53-54, 73]. If no infected users can recover after a worm attack, it is also called the Susceptible-Infectious (*SI*) model [6, 20-21]. Satorras and Vespignani [53] presented a differential equation for their *SIS* model by differentiating the infection dynamics of nodes with different degrees. Later, Moreno *et al.* [54-55] and Boguna *et al.* [34] provided a differential equation *SIR* model to study the dynamics of epidemic spreading on topological networks. As shown in Zou *et al.* work [6], such differential equations significantly overestimate the epidemic spreading speed due to their implicit homogeneous mixing assumption. In actual fact, the spreading of email worms is directly related to network topology. In our work, we avoid traditional overestimation problems (homogenous mixing) by examining the individual spreading steps and state transitions on each node in the network (see Section 6.3.2).

Zou *et al.* [6] also presented a simulation model on the propagation of email worms. Their paper demonstrates a fairly comprehensive analysis on the impact of various parameters, different topologies and selective percolation. However, some assumptions are not realistic. For example, the authors believe that just one malicious email copy will be sent to recipients even if an infected user checks multiple emails containing worms. In fact, only one malicious copy is sent whenever the infected user opens a reinfection worm email. We analyze and discuss these problems in Section 6.3.2 and Section 6.4.

In the work of Chen and Ji [32], a spatial-temporal random process was used to describe the statistical dependence of malware propagation in arbitrary topologies. However, there are

also some weak assumptions made. Firstly, [32] uses a *SIS* model, even though infected users are not likely to be infected again after they clean their computers by patching vulnerabilities or updating anti-virus software. Secondly, their model assumes that an infected computer cannot be reinfected. As we stated above, recent email worms are apt to reinfect users, which are far more aggressive in spreading throughout the network. Thirdly, the authors ignore an important human behavior; the email checking time, which has been proven to greatly affect the propagation of email worms. In this chapter, we also discuss the spatial and temporal processes in the propagation of worms, but we extend this and focus on more realistic reinfection email worms. Moreover, we synchronize the worms spreading time between nodes because of their different email checking time.

In recent years, there has also been some research on the propagation of isomorphic worms, such as Bluetooth worms, p2p worms [18-19], and worms on social networks [20]. Yan and Eidenbenz [21] presented a detailed analytical model that characterizes the propagation dynamics of Bluetooth worms. It captures not only the behavior of the Bluetooth protocol but also the impact of mobility patterns on the propagation of Bluetooth worms. However, all individual Bluetooth devices are homogeneously mixed, which overlooks the significant impact of topology. Fan and Xiang [19] used an ideal logic matrix to model the peer-to-peer propagation of worms. But in reality, their logic matrix is weak regarding an email resembling network because the weight of each link is a probability value ranging from zero to one instead of constant zero or one. Fan and Yeung [20] proposed a virus propagation model based on the application network of Facebook, which is the most popular among social network service providers. The difference between email worms and Facebook worms, as the authors highlight, is that people only check if there are any new emails and then log out while they spend more time on Facebook. Despite various differences among email worms and

other isomorphic worms, the manner in which they are spread is similar. Our work, therefore, can help create a better understanding of such models.

There is another type of worm which propagates through the vulnerabilities in the entire IP space [36, 78-79]. However, this propagation is unrelated to topology information and is already beyond the scope of this chapter. Our major focus is to understand the complex propagation dynamics of email worms, and thus, we focus solely on *SI* models and do not consider the recovery process.

## 6.3 Generality of the Propagation Model

### 6.3.1 Propagation Parameters

#### *6.3.1.1 Node Status* X*(t)*

Each node in the network has two different states: '*healthy*' and '*infected*'. 'Healthy' means the node is still susceptible and 'infected' means the node has been infected by email worms. We draw a basic state transition graph of an email user in Fig. 6.1(a). Moreover, an infected node sends out malicious emails at the precise moment when a user opens the worm email. Later, the node remains infected yet dormant until the process of disseminating malicious emails is triggered again. To facilitate the description, we set 'infected' as having two sub-states in terms of '*active*' and '*inactive*' respectively to denote an infected computer being at the stage of disseminating infectious emails or staying dormant. Let random variable $X_i(t)$ denote the status of a network node $i$ at discrete time tick $t$, so we have

$$X_i(t) = \begin{cases} 0 & \text{healthy} \\ 1 & \text{infected} \begin{cases} 1.1 & \text{active} \\ 1.2 & \text{inactive} \end{cases} \end{cases} \tag{6.1}$$

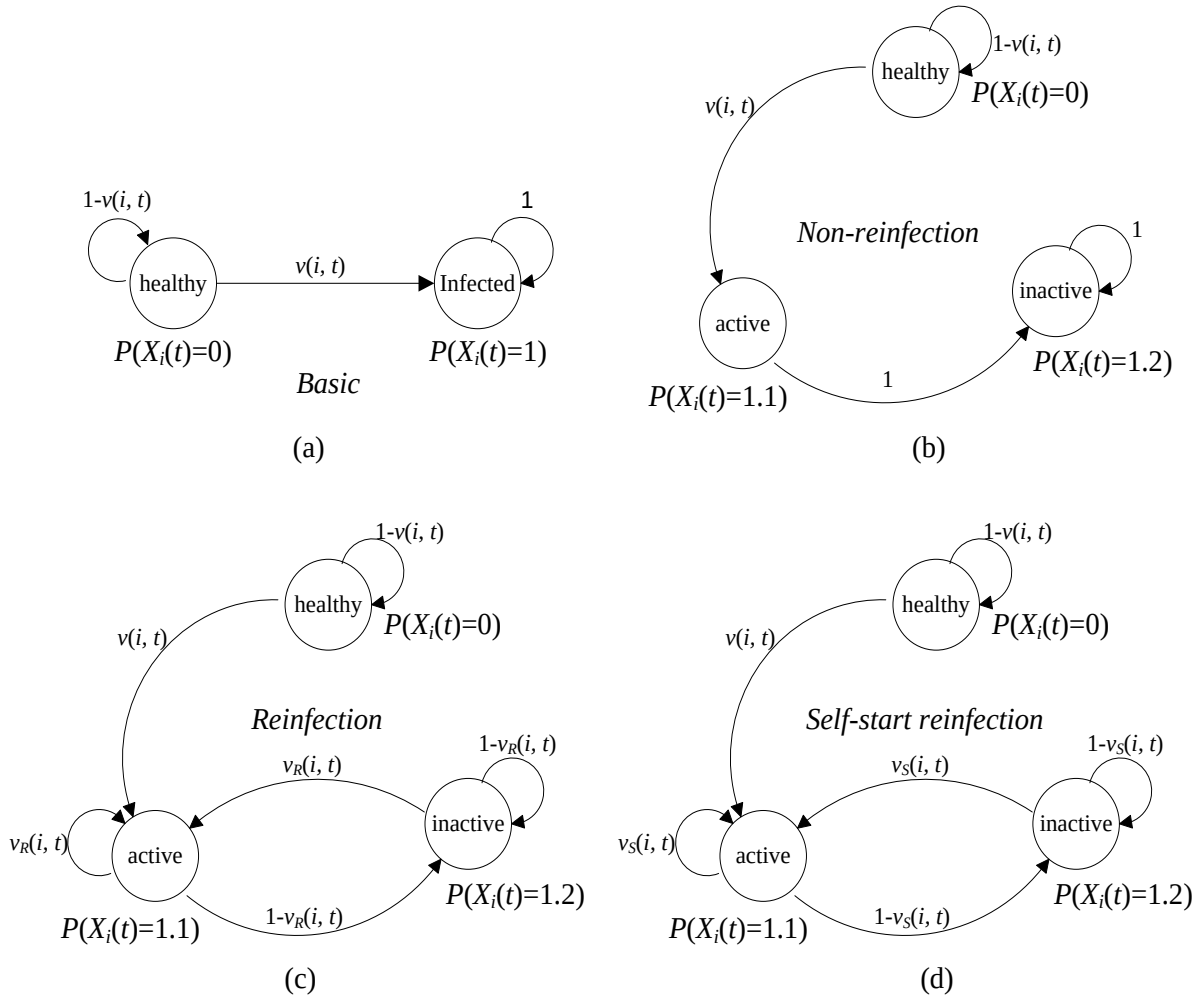



Figure 6.1: State transition graphs of an email user.

Our model consists of the propagation process at each discrete time tick. Each time tick $t$ can represent an arbitrary time interval in the real world, such as one minute, ten minutes, or even one hour. Thus, the absolute time tick value used in a discrete-time model does not matter, such as the mean value $E[CT_i]$=40 used in our model. On the other hand, since all events are assumed to happen right at discrete time ticks, a discrete-time model would be more accurate if a discrete time tick represents a shorter time interval. The expected number of infected nodes at time $t$, $n(t)$, can be easily computed from $P(X_i(t))$=1.

$$n(t) = E\left[\sum_{i=1}^{M} X_i(t)\right] = \sum_{i=1}^{M} E\left[X_i(t)\right] = \sum_{i=1}^{M} P\left(X_i(t) = 1\right) \tag{6.2}$$

#### *6.3.1.2 Email Checking Time (*CT*)*

The infection time includes network latency from node *i* sending a malicious email to node *j*, and an email checking time delay which is the time when a user opens a malicious email. Compared with time costs of checking an email, network latency can be ignored. In this chapter, we assume a worm copy is sent at time *t* and it will appear directly in the receiver's mailbox next time tick *t*+1. In fact, the email checking time of a user is a stochastic variable determined by the user's habits. For example, some users check email once every morning. Some users use email client programs to fetch and check email at a specified time interval or at a random time. We use $CT_i$ to denote the average email checking time period of node *i*. Each user *i* will check and read emails with their own $CT_i$. We use a random variable $open_i(t)$ to indicate the event of user *i* to check their mailbox at time *t*, as in

$$open_i(t) = \begin{cases} 0 & \text{user i does not check mailbox at time t} \\ 1 & \text{user i checks mailbox at time t} \end{cases} \tag{6.3}$$

To facilitate the description, we introduce $G(i, t)$ to indicate whether user *i* checks their mailbox at time *t* or not. This can help synchronize propagation dynamics between the nodes in the network, as in

$$G(i,t) = \begin{cases} 0 & \text{otherwise} \\ 1 & \text{t mod } \mathrm{CT}_i = 0 \end{cases} \tag{6.4}$$

When $G(i, t)$ is equal to one, user *i* is checking their mailbox. Therefore, we have the expression:

$$P\left(open_i(t) = 1\right) = G(i,t) \tag{6.5}$$

#### *6.3.1.3 Self-Start Time (*RT*)*

Currently, certain email worms, such as Win32/Mydoom and Win32.Imsolk, register themselves in start-up services and spread at every opportunity, and do not solely rely on a user opening emails. This kind of worm will automatically send out malicious copies once the system starts or when specific events are triggered. In this chapter, we employ $RT_i$ to represent the average self-start period of user *i*. Similar to the definition of email checking time *CT*, we also use the random variable $start_i(t)$ to indicate the 'self-start' event in user *i*'s computer at time *t*, as in

$$start_i(t) = \begin{cases} 0 & \text{system i does not start up at time t} \\ 1 & \text{system i start up at time t} \end{cases} \tag{6.6}$$

We also introduce $Q(i, t)$ to indicate whether the infected computer will send out malicious emails by the self-start process at time *t* or not, as in

$$Q(i,t) = \begin{cases} 0 & \text{otherwise} \\ 1 & \text{t mod } \mathrm{RT}_i = 0 \end{cases} \tag{6.7}$$

Similarly, we have the expression:

$$P\left(start_i(t) = 1\right) = Q(i,t) \tag{6.8}$$

#### *6.3.1.4 Propagation Matrix (*P*)*

Whether or not a computer can be infected by worm emails is determined by human factors, such as the user's personal habits of checking emails and their security consciousness. In our analytical model, we propose employing an *M* by *M* square matrix *P* with elements $p_{ij}$ to describe a network consisting of *M* nodes, wherein $p_{ij}$ represents the propagation probability of the worm spreading from user *i* to user *j*. Specifically, when the value of $p_{ij}$ is ***not*** equal to zero, it means the probability that user *j* is infected by opening malicious emails

received from user *i*. Otherwise, when $p_{ij}$ is equal to zero, it means there is no contact between user *j* and user *i*. Thus, matrix *M* also reflects the topology of an email network. We call this matrix the propagation matrix *P* of a network, as in

$$P = \begin{pmatrix} p_{11} & \cdots & \cdots \\ \cdots & p_{ij} & \cdots \\ \cdots & \cdots & p_{MM} \end{pmatrix}_{M \times M} \qquad p_{ij} \in [0,1] \tag{6.9}$$

If user *i* is susceptible, it can be compromised by any of its infected neighbors once this user opens a worm email. As shown in Fig. 6.1(b), for non-reinfection email worms, user *i* is susceptible only when this user is at a healthy stage. We have:

$$P\left(X_j(t) = 1.1 \mid X_i(t-1) = 1.1, X_j(t-1) = 0, open_j(t) = 1\right) = p_{ij} \tag{6.10}$$

However, as shown in Fig. 6.1(c), for reinfection email worms, user *i* is susceptible, not only at healthy but also at an active and inactive state. In addition to (6.11), we have:

$$P\left(X_j(t) = 1.1 \mid X_i(t-1) = 1.1, X_j(t-1) = 1, open_j(t) = 1\right) = p_{ij} \tag{6.11}$$

Moreover, as shown in Fig. 6.1(d), similar to reinfection worms, self-start reinfection email worms can drive an infected user to the 'active' state when the user restarts the computer or a specific event is triggered. Thus, in addition to (6.10) and (6.11), the propagation probability from user *i* to user *j* by worm emails but not the self-start process is as follows:

$$P\left(X_j(t) = 1.1 \mid X_i(t-1) = 1.1, start_j(t-1) = 0, open_j(t) = 1\right) = p_{ij} \tag{6.12}$$

### 6.3.2 Basic Analytical Model of the Propagation of Email Worms

According to (6.2), the expected number of infected users is ascribed to the sum of probability of being infected for each node in the network. Therefore, the following

discussion will be based on how to compute the probability of being infected for each node. The procedure of infection for each node can be expressed by a *state-transition graph* as shown in Fig. 6.1. When a healthy but susceptible user opens a worm email, it is infected immediately and the worm begins to search the local email contact book to send malicious copies. In this phase, this infected user is at the stage of 'active'. After this infected user sends out email copies to their friends, it transfers to the next step definitely, called the 'inactive' state, which means the node will not spread worms even if this infected user opens malicious copies again. We then have the following computation for the infected probability of each node at time *t*:

$$P(X_i(t)=1)=\underbrace{P(X_i(t-1)=1)}_{f1}+\underbrace{P(X_i(t)=1.1\,|\,X_i(t-1)=0)}_{f2}\underbrace{P(X_i(t-1)=0)}_{f3} \quad (6.13)$$

In (6.13), $f1$ and $f3$ can be iterated by difference equations. The problem becomes how to compute $f2$. To facilitate the description, we use $v(i,t)$ to represent $f2$. It indicates user $i$ is healthy at time $t$-1, but is infected at time $t$. If user $i$ does not open worm emails at time $t$, $v(i,t)$ is equal to zero. Therefore, we have:

$$\begin{aligned} &v(i,t)\\ &=\underbrace{P\left(X_i(t)=1.1, open_i(t)=0\,|\,X_i(t-1)=0\right)}_{\text{impossible event}}+P\left(X_i(t)=1.1, open_i(t)=1\,|\,X_i(t-1)=0\right)\\ &=P\left(X_i(t)=1.1, open_i(t)=1\,|\,X_i(t-1)=0\right) \end{aligned} \quad (6.14)$$

There is no relation between a user $i$ opening a worm email at time $t$ and whether this user is infected or not at time $t$-1. Therefore, the random events $X_i(t\text{-}1)$ and $open_i(t)$ are independent. According to the *theorem* 1, we can compute $v(i,t)$ as follows:

$$\begin{aligned} &v(i,t)\\ &=P\left(X_i(t)=1.1\,|\,X_i(t-1)=0, open_i(t)=1\right)P(open_i(t)=1)\\ &=P\left(X_i(t)=1.1\,|\,X_i(t-1)=0, open_i(t)=1\right)G(i,t) \end{aligned} \quad (6.15)$$

*Theorem* 1: we assume there are three arbitrary random variables: *A*, *B*, *C*. When *B* and *C* are independent, we have $P(AB|C)=P(A|BC)P(B)$.

*Proof*:

$$\begin{aligned}&\text{left} = P(ABC)/P(C) = P(ABC)P(B)/\big(P(B)P(C)\big)\\&= P(ABC)P(B)/P(BC) = \text{right}\end{aligned}$$

In the real world, users will check and read emails according to their own personal habits. Once users read malicious emails, worm copies are then sent out. In this chapter, we assume email users check their mailbox periodically. Thus, malicious emails in a user's mailbox may arrive at different times, though they will be read at the same time when the user visits their mailbox. We introduce *t'* to indicate an arbitrary time within a time period between when a user last checks their email and the current time *t* (excluding time *t*). It is significant to ascertain the number of unread emails after a user last checks their emails. We then have (as shown in Fig. 6.2):

$$\begin{cases} t - CT_i \le t' < t & \text{if G(i,t)=1} \\ t - (t \bmod CT_i) \le t' < t & \text{otherwise} \end{cases} \tag{6.16}$$

In (6.15), we use $s(i,t)$ to represent $P(X_i(t)=1.1|X_i(t\text{-}1)=0, open_i(t)=1)$. Different from $v(i,t)$, this indicates the probability of user *i* being healthy at time *t*-1 but infected at time *t* under the condition that the user opens the mailbox at time *t*. Let $N_i$ denote all neighbors of node *i*, $N_i = \{j|p_{ij}\neq 0, \forall j\}$. The malicious emails in a user's mailbox come from neighbors $N_i$. As a result, we have the following computation:

$$\begin{aligned}&s(i,t)\\&= P\left(X_i(t)=1.1, \exists j \in N_i \ X_j(t') \neq 1.1 \middle| X_i(t-1)=0, open_i(t)=1\right) +\\&\quad \underbrace{P\left(X_i(t)=1.1, \forall j \in N_i \ X_j(t') \neq 1.1 \middle| X_i(t-1)=0, open_i(t)=1\right)}_{\text{impossible event}}\\&= P\left(X_i(t)=1.1, \exists j \in N_i \ X_j(t') \neq 1.1 \middle| X_i(t-1)=0, open_i(t)=1\right)\end{aligned} \tag{6.17}$$

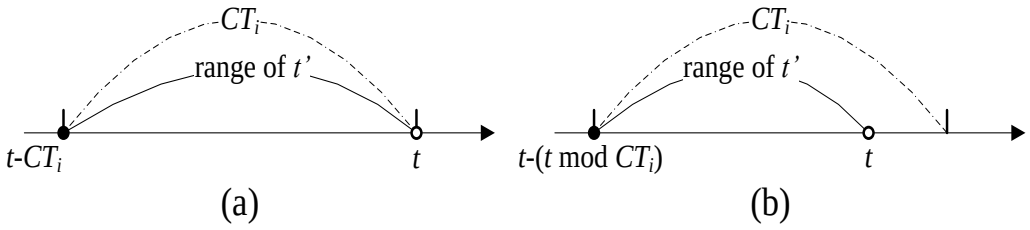



Figure 6.2: Different cases of the parameter *t'*.

If user *i* receives worm emails from its neighbors, then the probability for user *i* to be infected is as follows:

$$s(i,t) = 1 - \prod_{j \in N_i} \left[ 1 - P\left( X_i(t) = 1.1, X_j(t') = 1.1 \middle| X_i(t-1) = 0, open_i(t) = 1 \right) \right] \tag{6.18}$$

In (6.18), the events $X_j(t')=1.1$ and $X_i(t\text{-}1)=0$ are dependent [32]. According to our investigation [80], the dependence of the above events is mainly caused by the cycles in the propagation procedure. However, it is really a challenge to estimate the effect of this dependence. The conditional probability $P(X_j(t')=1.1|X_i(t\text{-}1)=0)$ is computationally too expensive to obtain, especially when the size of a neighborhood is large. In paper [32], the authors use two approximations for modeling a worm's propagation. Readers can find extensive discussion in [32, 80]. In this chapter, we use the simple approximation from [32] and consider they are independent. We then have:

$$\begin{aligned} & s(i,t) \\ & = 1 - \prod_{j \in N_i} \Big[ 1 - P(X_i(t) = 1.1 \,|\, X_j(t') = 1.1, X_i(t-1) = 0, \\ & \quad open_i(t) = 1) P(X_j(t') = 1.1) \Big] \\ & = 1 - \prod_{j \in N_i} [1 - p_{ji} \underbrace{P(X_j(t') = 1.1)}_{f1}] \end{aligned} \tag{6.19}$$

In this section, we have elaborated a basic propagation modeling mechanism. By using difference equations to iterate function *s*(*i*,*t*), we are able to estimate the number of infected

nodes in the network at time *t*. In the following sections, we will derive the computation of *f*1 of (6.19) in different cases respectively, because it has different values by different kinds of email worms.

## 6.4 Modeling of Non-reinfection Email Worms

### 6.4.1 How Non-reinfection Worms Work

Non-reinfection email worms usually appear early on. The state transition graph of an email user is shown in Fig. 6.1(b). For each 'active' user, he sends out one malicious copy only once even if worm emails are checked several times. Subsequently, the infected user becomes 'inactive' and stays dormant during propagation.

A healthy email user will be infected only when its neighbors are in the active stage. In Fig. 6.3, we set up a simple example of non-reinfection worms spreading in three time ticks. User 1, 2 and 3 are a victim's neighbors. In Fig. 6.3(a), user 1 and user 2 check their mailboxes and are infected.

At the same time, two malicious copies are sent to the victim. This process happens at the time *t*-2. In Fig. 6.3(b-1), user 3 checks their mailbox and reads both emails. As a result, a copy of the worm is sent to the victim at time *t*-1. At this time, there are already two worm copies in the victim's mailbox. Then, in Fig. 6.3(c-1), the victim receives a total of three worm copies from their neighbors at time *t*.

By investigating the above scenario, we derive the spreading nature of non-reinfection email worms: 1) an infected email user has and only has one chance to spread worm copies in an active state; 2) once being infected, an email user will send out just one copy to their friends. Non-reinfection email worms spread less efficiently in a network.

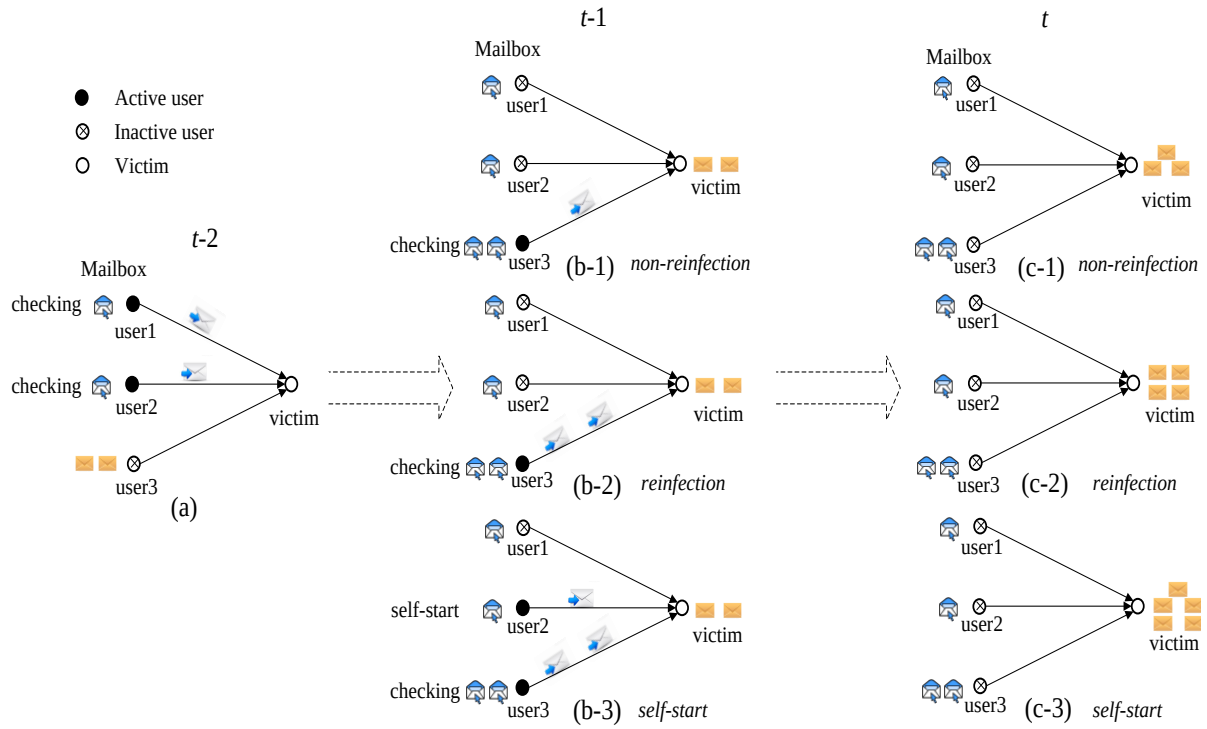



Figure 6.3: Example of email worms spreading between nodes in the network.

## 6.4.2 The Model

In order to model the propagation of non-reinfection email worms, we extend $f1$ in (6.19). If a user $j$ is in an active stage at time $t'$, according to Fig. 6.1(b), it should be healthy at time $t'$-1. Then we have:

$$s(i,t) = 1 - \prod_{j \in N_i} \left[ 1 - p_{ji} v(j, t'-1) P(X_j(t'-1) = 0) \right] \tag{6.20}$$

We disassemble (6.20) and then we have:

$$s(i,t) = 1 - \prod_{j \in N_i} [1 - p_{ji} \underbrace{v(j, t-2) P(X_j(t-2) = 0)}_{f1}] \underbrace{\prod_{j \in N_i, t' \neq t-1} \left[ 1 - p_{ji} v(j, t'-1) P(X_j(t'-1) = 0) \right]}_{f2} \tag{6.21}$$

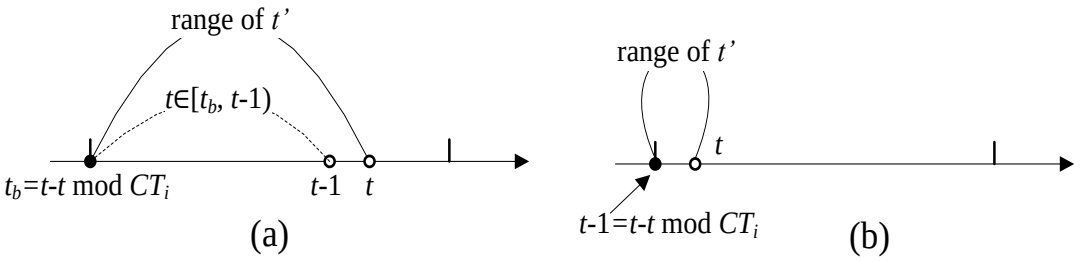



Figure 6.4: Two cases in the iteration of $s(i,t)$

In (6.2), $f1$can be iterated by difference equations. $f2$ is similar to (6.21) except that $f2$ excludes the infection process at time $t$-1. According to the condition that if user $i$ checks their mailbox at time $t$-1, we have time $t'$ drop in a range from the time user $i$ last checked the mailbox to $t$-1 as shown in Fig. 6.4(a). Therefore, $f2$ in (6.21) is equivalent to $s(i, t\text{-}1)$. $f2$ in this case, records the effect of emails received from the time user $i$ last checked their mailbox to $t$-1. Then we have:

$$s(i,t) = 1-\big(1-s(i,t-1)\big)\prod_{j\in N_i}\Big[1-p_{ji}v(j,t-2)P(X_j(t-2)=0)\Big] \tag{6.22}$$

However, as shown in Fig. 6.4(b), if user $i$ checked their mailbox at time $t$-1, the mailbox at current time $t$ would only contain emails sent from neighbors at time $t$-1. Thus, $f2$ is meaningless and equal to one. So we have

$$s(i,t) = 1-\prod_{j\in N_i}\Big[1-p_{ji}v(j,t-2)P(X_j(t-2)=0)\Big] \tag{6.23}$$

We unify the two cases of a user checking mailbox at time $t$-1, then we have:

$$\begin{aligned} s(i,t) = 1-[1-&\underbrace{G(i,t-1)}_{f3}\underbrace{s(i,t-1)}_{f4}] \\ &\prod_{j\in N_i}[1-p_{ji}\underbrace{v(j,t-2)}_{f1}\underbrace{P(X_j(t-2)=0)}_{f2}] \end{aligned} \tag{6.24}$$

In (6.24), $f1$, $f2$, $f3$ and $f4$ represent variables of user $i$ at time $t$-1. Therefore, the infected number ($n(t)$) in the network at time $t$ can be iterated step by step through the difference equations of (6.2), (6.15) and (6.24).

### 6.4.3 Evaluation of the Non-reinfection Email Worms Model

We represent the topology of the logical email network by a directed graph as the sending and receiving of emails is governed by different processes. A widely-studied typical complex network [6, 32] has a power-law topology, where the nodal degree distribution is characterized as $P(k) \sim k^{-\alpha}$ with $P(k)$ being the probability that a node has a degree of $k$ [47, 81]. We choose a simple power-law network generator proposed in Chapter 5 (See Section 5.3.1.1) instead of other generators because it has an adjustable power-law exponent $\alpha$. Paper [34] refers to another concept in email networks: the correlated or uncorrelated email network. For a correlated email network, there is a heightened chance that an email user will have some people in their contact list if they have this person in theirs. Besides, the email addresses of individuals who have large address books tend to appear in the address books of many others. In this chapter, we use *reciprocity* to indicate the fraction of edges between users that point both ways and follow the research findings of [62]: the *reciprocity* is equal to 0.23.

We compare the performance of our proposed analytical model with that of some well-known models: the simulation model [6] and the spatial-temporal model [32]. These two models have been verified by the authors to be more accurate than earlier models, such as the epidemic models in [34, 53-55] and the AAWP model [10]. In this chapter, we expect the evaluation of our model to be closer to the simulation model than the spatial-temporal model.

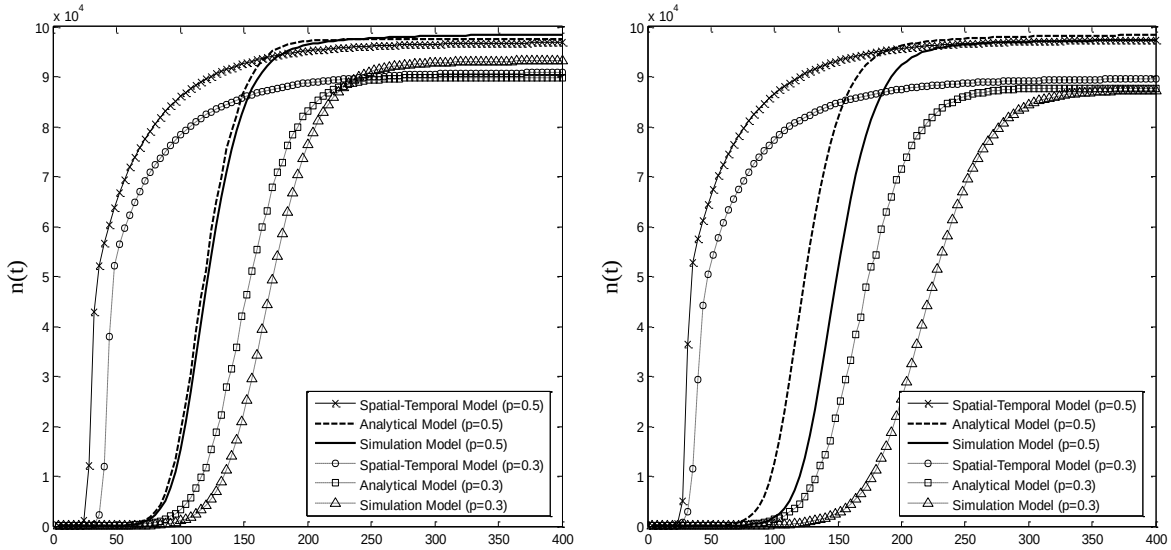



Figure 6.5: The propagation of non-reinfection worms with different infection probability *p*. (a) Uncorrelated network with $CT \sim N(40, 20^2)$; (b) Correlated network with $CT \sim N(40, 20^2)$

It should be noted that we compare our model with the independent spatial-temporal model according to the independence assumption in this chapter.

Our implementation is in Visual C++ 2008 SP1 and Matlab 7. The random numbers in our experiments are produced by the C++ TR1 library extensions. In the experiments, we set two initially infectious nodes. The degree of the topology follows the power-law distribution ($\alpha$=2.58). We assume the total number of nodes in the network is 100,000, and the simulation program runs 100 times.

In this subsection, we carry out two experiments to evaluate the performance of the propagation of non-reinfection worms with the same parameters as [6]. Fig. 6.5 shows the comparison of the aforementioned models in the uncorrelated and correlated network respectively, with infection probability $p$=0.5 or $p$=0.3 and $CT \sim N(40, 20^2)$. Firstly, we found the performance of our model much closer to the simulation model than to the spatial-temporal model. The main reason is that our analytical model takes into account the user checking period, whereas the spatial-temporal model considers that a user opens an email at the time of receiving it. Apparently, this is not realistic according to the above statement.

Thus, the spreading speed of the spatial-temporal model is faster and cannot match the simulation model very well. Secondly, the analytical model in Fig. 6.5(a) fits the simulation model better than Fig. 6.5(b). This is because the *dependence* effect on the uncorrelated network is weaker than that on the correlated network. We have proven this as follows.

*Proof*: For an uncorrelated email network, the effect on the dependence of events (e.g. $X_j(t)$=1.1and $X_i(t\text{-}1)$=0) is weak. We generate the uncorrelated email network as follows. For each user, we let a user's out-bound edge point to any other randomly selected users, and thus the out-degree and in-degree of users is uncorrelated. According to our investigation [80], the dependence of the above events is mainly caused by propagation cycles in the spreading procedure. A propagation cycle is a spreading route from one user back to itself through several intermediate users. For a cycle with one intermediate user, we have:

$$P\left(x_i(t)=1.1 \mid x_i(t-2CT_i)=1.1\right) = \frac{D}{N-1}\bullet\frac{D}{N-1}\bullet D\bullet E^2\left(p_{ij}\right)=\frac{D^3}{\left(N-1\right)^2}\bullet E^2\left(p_{ij}\right)$$

wherein, $D$ is an average out-degree of each user. $N$ is the network size. $E(p_{ij})$ is the mean of the propagation probability of the worm spreading from any user $i$ to user $j$. Generally, for the cycle with $k$ intermediate users, we then have:

$$P\left(x_i(t)=1.1 \mid x_i\left(t-(k+1)CT_i\right)=1.1\right) = \left(\frac{D}{N-1}\right)^{k+1}\bullet D^k\bullet E^{k+1}\left(p_{ij}\right)=\left(\frac{D^2}{N-1}\right)^{k+1}\bullet\frac{1}{D}\bullet E^{k+1}\left(p_{ij}\right)$$

In this chapter, we generate the email network with $D$=8, $E(p_{ij})$=0.5 and $N$=100,000. It is easy to achieve the result where the above formula has a maximum when $k$ is equal to one. Then we have:

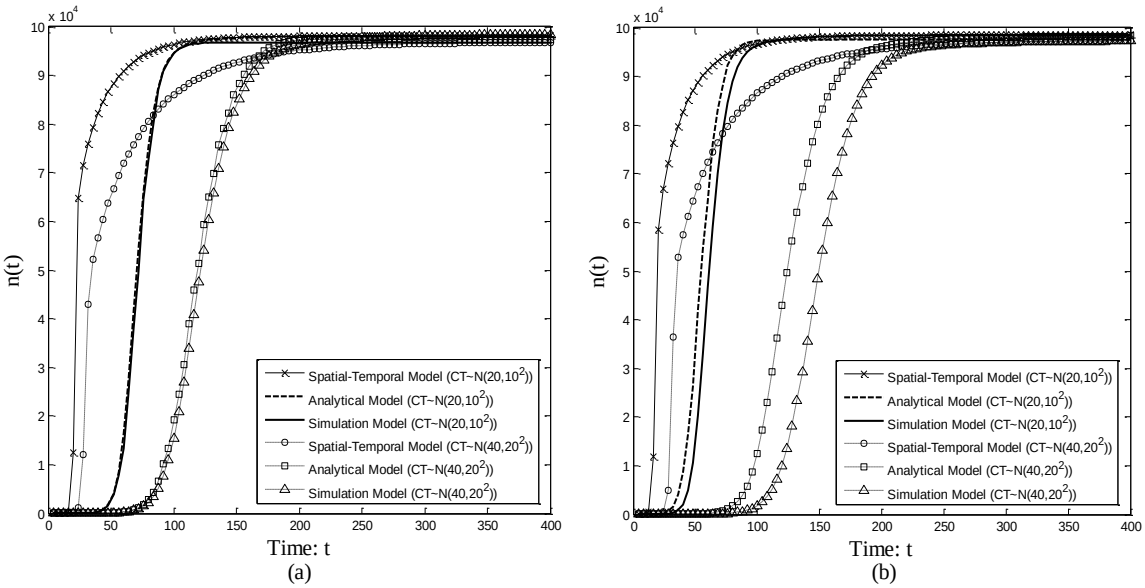
$$P\left(x_i(t)=1.1 \mid x_i\left(t-(k+1)CT_i\right)=1.1\right) \le \frac{D^3}{\left(N-1\right)^2}E^2\left(p_{ij}\right)=\frac{8^3}{\left(100000-1\right)^2}0.5^2=1.28\times10^{-8}$$

Therefore, we prove that the negative effect on dependence is weak in the uncorrelated network.

Nevertheless, due to the correlated network having a large *reciprocity* (0.23), many propagation cycles exist in the spreading procedure. As discussed in [80], a large number of cycles lead to the non-negligible dependence effect on the correlated network. Moreover, it is observed that the infection probability $p$ can affect the accuracy of the model. In both Fig. 6.5(a) and (b), the analytical model with a larger infection probability ($p$=0.5) fits better with the simulation model than the case with a smaller infection probability ($p$=0.3). In fact, the infected scale and speed are largely determined by the early stage of the worm's propagation [79]. If the infection probability is small, the propagation procedure in the simulation model can probably be stopped in the early stages. This results in the small infected scale of the network greatly reducing the mean value of $n(t)$. We then have:

*Remark* 1: Our analytical model performs better when the email worm is more deceptive (means larger infection probability).

Different from Fig. 6.5, Fig. 6.6 compares models in the uncorrelated and correlated network respectively with different email checking time $CT \sim N(40,20^2)$ or $CT \sim N(20,10^2)$ and $p$=0.5. From both Fig. 6.6(a) and (b), we can conclude that the spreading speed is faster if the email checking period is shorter. Similar to Fig. 6.5, when compared with models in an uncorrelated network, the models in a correlated network have a larger *dependence* effect on the propagation procedure. In the real world, the correlated network is more realistic [62]. Thus, we have:




(a) (b)

Figure 6.6: The propagation of non-reinfection worms with different email checking time *CT*. (a) Uncorrelated network with $p$=0.5; (b) Correlated network with $p$=0.5

*Remark* 2: We have to improve the accuracy of our analytical model by eliminating the *dependence* effect. An inspired measurement is to integrate the Markov approximation [32] or propagation cycles [80] into our analytical model.

## 6.5 Modeling of Reinfection Email Worms

### 6.5.1 How Reinfection Worms Work

Reinfection email worms can greatly accelerate the worm spreading speed as the malicious copy will be sent out every time the user opens the worm email. The state transition graph of an email user is shown in Fig. 6.1(c). Different from non-reinfection, when an inactive user checks a worm email, this user will become active and once again send out malicious copies to their neighbors. In order to investigate the propagation process among the email users in the network, we have another example of reinfection worms spreading in three time ticks in Fig. 6.3. Similar to non-reinfection email worms, in Fig. 6.3(a), infected user 1 and user 2 send two malicious copies to the victim at time $t$-2. In Fig. 6.3(b-2), user 3 reads both emails

inside the mailbox and two copies of the worm are sent to the victim at time *t*-1. As a result, in Fig. 6.3(c-2), the victim receives a total of four worm copies from their neighbors at time *t*.

Through analysis of the individual steps and the state transition above, we derive the spreading nature of reinfection email worms as follows: 1) an infected user will go into an active state and send out worm copies after being infected not only from a healthy state but also from an infected state; 2) the number of malicious emails sent by an infected user is determined by the number of worm emails this user reads when they open their mailbox. Compared with non-reinfection worms, reinfection email worms are far more efficient to spread in a network.

### 6.5.2 Underestimation in the Traditional Simulation Model

The traditional simulation model [6] ignores the second part of propagating reinfection worms. An infected user in the simulation model always sends only one worm copy to their neighbors even if the user opens two or more infectious emails. Take Fig. 6.3(c-2) for example, one of the two emails from user 3 will be neglected and the victim will receive a total of three emails at time *t*.

However, the problem is not as simple as we have discussed above. If we revise the simulation model to satisfy the second spreading nature, the simulation model becomes a time-consuming process. The situation becomes worse when the scale of the email network is enlarged. In Fig. 6.7, we observe that the number of emails received by each user increases exponentially. Intuitively, this phenomenon is attributed to the *snowball* effect: we simply suppose each email user is connected to *m* users and will read all emails received. Initially, each user has *m* worm emails. Subsequently, users will receive $m^2$ worm emails after one

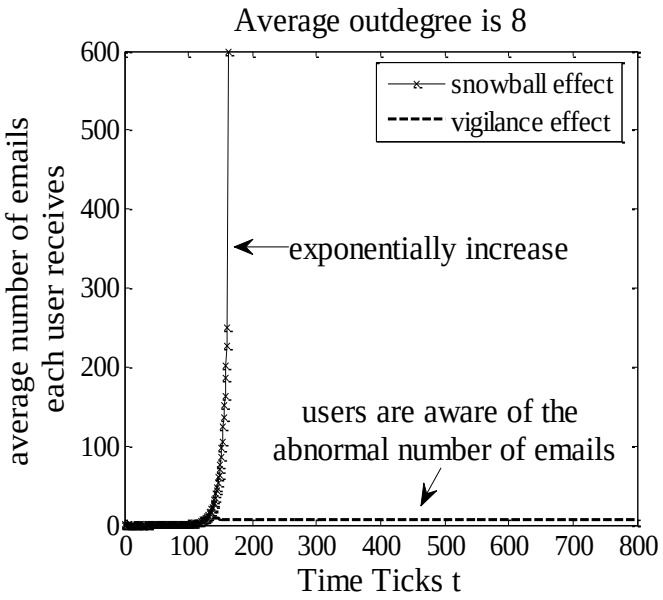



Figure 6.7: Snowball effect and vigilance effect.

mailbox checking period. Similarly, it is easy to know that each user will be overwhelmed by worm emails within a short period of time.

In the real world, however, email recipients may be aware of the number of emails they receive in their mailbox. If the number of emails exceeds the usual number, email users may not open all of them. A user's vigilance leads to an infected user sending out more than one but still a limited number of worm copies to their friends, which is mainly determined by social engineering techniques the worm adopts and also the user's awareness. For example, email worms like Mydoom [72] have many different subject topics. Some email worms, like w32.Imsolk [58], are more deceiving because the email titles are labeled with "Here you are". The *vigilance effect* is hard to estimate and we do not know the true impact of the real worms propagation. In order to see how it affects the spreading procedure, we introduce a vigilance degree as $\beta$. In this chapter, we assume email users mainly communicate with their friends and the number of emails sent by users will not exceed $\beta$ times the length of their contact list. We will evaluate the *vigilance effect* with a series of $\beta$ (See Section 6.5.5).

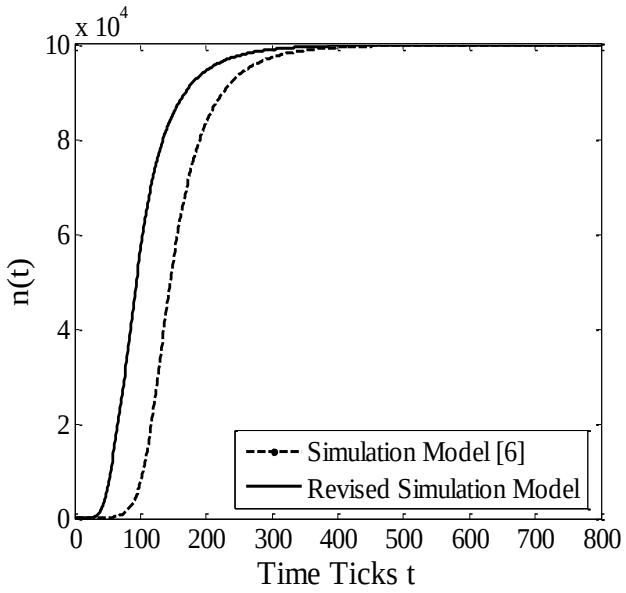



Figure 6.8: Underestimation in the traditional simulation model.

According to the above analysis, we revise the simulation model [6] and compare it. As shown in Fig. 6.8, it is noticeable that the simulation model significantly underestimates the spreading ability of reinfection email worms. Later, we will use the revised simulation model to evaluate our analytical model.

### 6.5.3 Virtual User

We use six nodes to illustrate the propagation between email users in the network. As shown in Fig. 6.9(a), User $U_5$ may be infected three times by possibly opening one to a maximum of three malicious email attachments from $U_1$, $U_2$ and $U_3$. User $U_6$ receives emails from $U_4$ and $U_5$ but may be infected again by the possible arrival of another two malicious emails from $U_5$. In order to model the infection process of $U_6$, we propose a concept of *virtual users* to explain the possible repetitious infection caused by $U_5$ opening more than one worm copies. As is shown in Fig. 6.9(b), $U_{5\_1}$ represents $U_6$ being infected by $U_5$ for the first

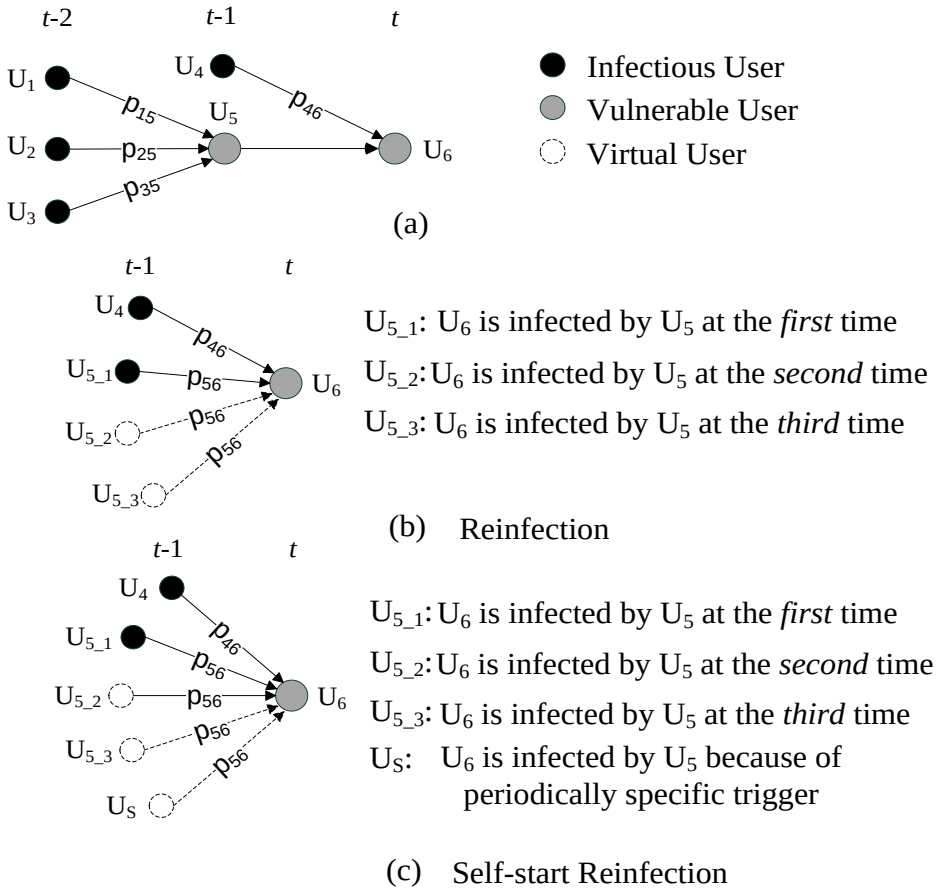



Figure 6.9: The propagation of reinfection and self-start reinfection worms.

time. If $U_5$ reads two worm emails, we use $U_{5\_2}$ to represent the possible infection of $U_6$ for the second time. Similarly, $U_{5\_3}$ represents the third possible infection if $U_5$ reads three emails. To facilitate the explanation, we simply set the email checking period as one time tick and the current time as $t$ in this example. $U_1$, $U_2$, $U_3$ and $U_4$ initially have only one worm email in their mailbox. Then we have:

$$P(X_{5\_1}(t-1)=1.1)=1-\prod_{j=1}^{3}\left[1-P(X_j(t-2)=1.1)\,p_{j5}\right] \tag{6.25}$$

We introduce a random variable $k_i(t)$ to denote the number of emails user $i$ reads at time $t$. Besides, we use variable $Y_{ij}(t)=P(X_i(t\text{-}1)=1.1)p_{ij}$ to indicate the probability of user $j$ having received and read the email from user $i$ and variable $\bar{Y}_{ij}(t)= 1\text{-}\ P(X_i(t\text{-}1)=1.1)p_{ij}$ to indicate the negation of $Y_{ij}(t)$. We have

$$\begin{aligned} &P\left(X_{5\_2}(t-1)=1.1\right) \\ &=P\left(X_{5\_1}(t-1)=1.1\right)-P\left(k_5(t-1)=1\right) \\ &=P\left(X_{5\_1}(t-1)=1.1\right)-\left(Y_{15}(t)\overline{Y_{25}(t)}\,\overline{Y_{35}(t)}+\overline{Y_{15}(t)}Y_{25}(t)\overline{Y_{35}(t)}+\overline{Y_{15}(t)}\,\overline{Y_{25}(t)}Y_{35}(t)\right) \end{aligned} \tag{6.26}$$

$$\begin{aligned} &P\left(X_{5\_3}(t-1)=1.1\right) \\ &=P\left(X_{5\_2}(t-1)=1.1\right)-P\left(k_5(t-1)=2\right) \\ &=P\left(X_{5\_2}(t-1)=1.1\right)-\left(Y_{15}(t)Y_{25}(t)\overline{Y_{35}(t)}+Y_{15}(t)\overline{Y_{25}(t)}Y_{35}(t)+\overline{Y_{15}(t)}Y_{25}(t)Y_{35}(t)\right) \end{aligned} \tag{6.27}$$

Actually, if the number of users in the network is large enough, it is exceptionally hard to examine $P(k_i(t\text{-}1)=K)$ on each $K$ by each user. $K$ is the number of emails that a user has opened. Suppose the network has $m$ users, the complexity of the algorithm to obtain the infection probability of each virtual user is $O(m^3)$. In this chapter, we adopt an approximate calculation. Generally, the Bernoulli experiment is widely used to model the number of successes in a sample drawn from a large population. Thus, we use the approximation as follows:

$$P\left(X_{5\_2}(t-1)=1.1\right)\approx P\left(X_{5\_1}(t-1)=1.1\right)-C_3^1 p_{5|ave}(t)\left(1-p_{5|ave}(t)\right)^2 \tag{6.28}$$

whereas $C_3^1$ is the Bernoulli coefficient. $P_{i|ave}(t)$ denotes the average probability of user $j$ having received and read the email from user $i$, as in

$$p_{5|ave}\left(t\right)=\frac{1}{3}\left(Y_{15}(t)+Y_{25}(t)+Y_{35}(t)\right) \tag{6.29}$$

We also have,

$$\begin{aligned} &P\left(X_{5\_3}(t-1)=1.1\right) \\ &\approx P\left(X_{5\_2}(t-1)=1.1\right)-C_3^2 p_{5|ave}^2(t)\left(1-p_{5|ave}(t)\right) \end{aligned} \tag{6.30}$$

Therefore, the probability of user $U_6$ being infected and sending out worm copies is as follows:

$$\begin{aligned} &P\left(X_6(t)=1.1\right) \\ &=1-\left(1-p\left(X_4(t-1)=1.1\right)p_{46}\right)\prod_{i=1}^{3}\left(1-p\left(X_{5-i}(t-1)=1.1\right)p_{56}\right) \end{aligned} \tag{6.31}$$

### 6.5.4 The Model

In this section, we follow the above discussion and example to build the propagation model of reinfection email worms. As shown in Fig. 6.1(c), not only a healthy but also an infected user can become active and send out worm emails to their neighbors. We define $v_R(i,t)$ as the probability of user $i$ having been infected at time $t$-1 and being active at time $t$. Then we have:

$$v_R\left(i,t\right)=P\left(X_i(t)=1.1\middle|X_i(t-1)=1\right) \tag{6.32}$$

In order to model the propagation of reinfection email worms, we extend $f1$ in (6.19) of the basic model. Similar to the non-reinfection model, we assume the events $X_j(t')$=1.1 and $X_i(t$-1)=1 are independent in this chapter. Then according to theorem 2, we have:

$$\begin{aligned} &s\left(i,t\right) \\ &=1-\prod_{j\in N_i,t'}\Big[1-p_{ji}\Big(v(j,t'-1)P\left(X_j(t'-1)=0\right)+ \\ &\quad v_R(j,t'-1)P\left(X_j(t'-1)=1\right)\Big)\Big] \\ &=1-\prod_{j\in N_i,t'}\left[1-p_{ji}v(j,t'-1)\right] \end{aligned} \tag{6.33}$$

*Theorem* 2: when the events $X_j(t')$=1.1 and $X_i(t$-1)=1 are independent, there is $v_R(i,t)=v(i,t)$.

*Proof*: similar to (6.15), (6.17) and (6.18), we have the following derivation:

$$\begin{aligned} &v_R\left(i,t\right)=P\left(X_i(t)=1.1\middle|open_i(t)=1,X_i(t-1)=1\right)G(i,t) \\ &=P\left(X_i(t)=1.1,\exists j\in N_i\ X_j(t')=1.1\middle|open_i(t)=1,X_i(t-1)=1\right)G(i,t) \\ &=\Big\{1-\prod_{j\in N_i,t'}\Big[1-P\left(X_i(t)=1.1,\right. \\ &\quad \left.X_j(t')=1.1\middle|open_i(t)=1,X_i(t-1)=1\right)\Big]\Big\}G(i,t) \end{aligned}$$

If the events $X_j(t')=1.1$ and $X_i(t\text{-}1)=1$ are independent, similar to (6.19), we then prove the proposition as in:

$$v_R(i,t)=\left[1-\prod_{j\in N_i,t'}\left(1-p_{ji}P\left(X_j(t')=1.1\right)\right)\right]G(i,t)=s(i,t)G(i,t)=v(i,t)$$

We continue to disassemble (33) for iteration, as in

$$s(i,t)=1-\prod_{j\in N_i,t'\neq t-1}\left[1-p_{ji}v(j,t'-1)\right]\prod_{j\in N_i}\left(1-p_{ji}v(j,t-2)\right) \tag{6.34}$$

According to the condition, if user $i$ checks their mailbox at time $t$-1, $s(i,t)$ may have different results. Similar to the analysis in Section 6.4.2, we have a unified conclusion as in

$$s(i,t)=1-\left(1-G(i,t-1)s(i,t-1)\right)\prod_{j\in N_i}\left(1-p_{ji}v(j,t-2)\right) \tag{6.35}$$

Different from non-reinfection email worms, the neighbors $N_i$ of user $i$ should be composed of two parts: real users and virtual users. We use $N_{i|R}$ to represent real neighbors and $N_{i|V}$ to represent virtual neighbors. Note that $N_{i|R}$ is constant for each user $i$ in our topology, but $N_{i|V}(t)$ is determined by the propagation procedure of worms. $N_{i|V}(t)$ varies at different time $t$. Thus, we have

$$s(i,t)=1-\left(1-G(i,t-1)s(i,t-1)\right)\prod_{j\in N_{i|R}}\left(1-p_{ji}v(j,t-2)\right)\prod_{j\in N_{i|V}}\left(1-p_{ji}v(j,t-2)\right) \tag{6.36}$$

Visual users are created when corresponding real users are infected. We assume that visual users send worm copies to their neighbors at the time they are created. That is, visual users are healthy before, but infected when they are created. Thus, for a visual user which is created at time $t$, we have:

$$
\begin{aligned}
&P\left(X_j(t)=1.1\right) \quad (j \in N_{i|V}) \\
&= P\left(X_j(t)=1.1 \middle| X(t-1)=0\right) \underbrace{P\left(X_j(t-1)=0\right)}_{f1} + \\
&\quad \underbrace{P\left(X_j(t)=1.1 \middle| X(t-1)=1\right) P\left(X_j(t-1)=1\right)}_{f2} \\
&= P\left(X_j(t)=1.1 \middle| X(t-1)=0\right) \\
&= v(j,t)
\end{aligned}
\tag{6.37}
$$

Because visual users are supposed to be initially healthy, we have $f1$ is equal to one and $f2$ is equal to zero in (6.37). According to the analysis of the virtual users and the *vigilance effect* in Section 6.5.2 and 6.5.3, we have:

$$
\begin{aligned}
&s(i,t) \\
&= 1 - \underbrace{\left(1 - G(i,t-1)s(i,t-1)\right) \prod_{j \in N_{i|R}} \left(1 - p_{ji} v(j,t-2)\right)}_{=\lambda} \\
&\quad \prod_{j \in N_{i|V}(t-2)} \left(1 - p_{ji} P\left(X_j(t-2)=1.1\right)\right) \\
&= 1 - \lambda \prod_{j \in N_{i|V}(t-2)} \left(1 - p_{ji} P\left(X_{j\_K}(t-2)=1.1\right)\right)
\end{aligned}
\tag{6.38}
$$

wherein $||N_{i|V}(t)||$ is the number of visual users at time $t$, which can be iterated in the propagation procedure. In this chapter, we assume the number of emails sent by a user $i$ will not exceed $\beta$ times the length of user $i$'s contact list ($D_i$), so we use $min(||N_{i|V}(t)||, D_i)$ to obtain the minimum value. $K$ ranges from 1 to $min(||N_{i|V}(t-2)||, \beta D_i)$. $P(X_{j\_K}(t)=1.1)$ is the probability for the $k$-th virtual user of real user $j$ to be at an active state. We calculate $P(X_{j\_K}(t)=1.1)$ as in:

$$
\begin{aligned}
&P\left(X_{j\_K+1}(t)=1.1\right) \\
&= P\left(X_{j\_K}(t)=1.1\right) - P\left(k_j(t)=K\right) \\
&\approx P\left(X_{j\_K}(t)=1.1\right) - C^K_{||N_{i|V}(t-2)||} P^K_{5|ave} (1 - P_{i|ave})^{||N_{i|V}(t-2)||-K}
\end{aligned}
\tag{6.39}
$$

$P_{i|ave}$ can be calculated as in

$$
p_{i|ave}(t) = \frac{1}{||N_i||} \sum_{j \in N_i} Y_{ji}(t) \tag{6.40}
$$

In (6.38), all the components can be determined by the variables of user *i* at time *t*-1 or *t*-2. Therefore, the number of infected users in the network ($n(t)$) can be estimated by difference equations of (6.2), (6.15) and (6.38).

### 6.5.5 Evaluation of the Reinfection Email Worms Model

In this subsection, we carry out three experiments to evaluate the performance of propagation for reinfection worms with the same parameters as [6]. To the best of our knowledge, there are no analytical models that describe the propagation of reinfection and self-start reinfection worms. [32] discussed the propagation of worms in the network on the basis of a non-reinfection spreading mechanism. In this subsection, we compare our analytical model with the simulation model and expect its performance to be closer to the simulation model.

Fig. 6.10 shows the comparison of models in the uncorrelated and correlated network respectively with infection probability $p$=0.5 or $p$=0.3 and $CT \sim N(40, 20^2)$. Similar to the non-reinfection case, the results in the uncorrelated network are a lot better in relation to performance than in the correlated network. Meanwhile, it is observed that our analytical model is fairly accurate to the simulation model if users have a higher probability of opening malicious emails. Fig. 6.11 depicts the comparison of models in the uncorrelated and correlated network respectively with different email checking time $CT \sim N(40,20^2)$ or $CT \sim N(20,10^2)$ and $p$=0.5. We then have:

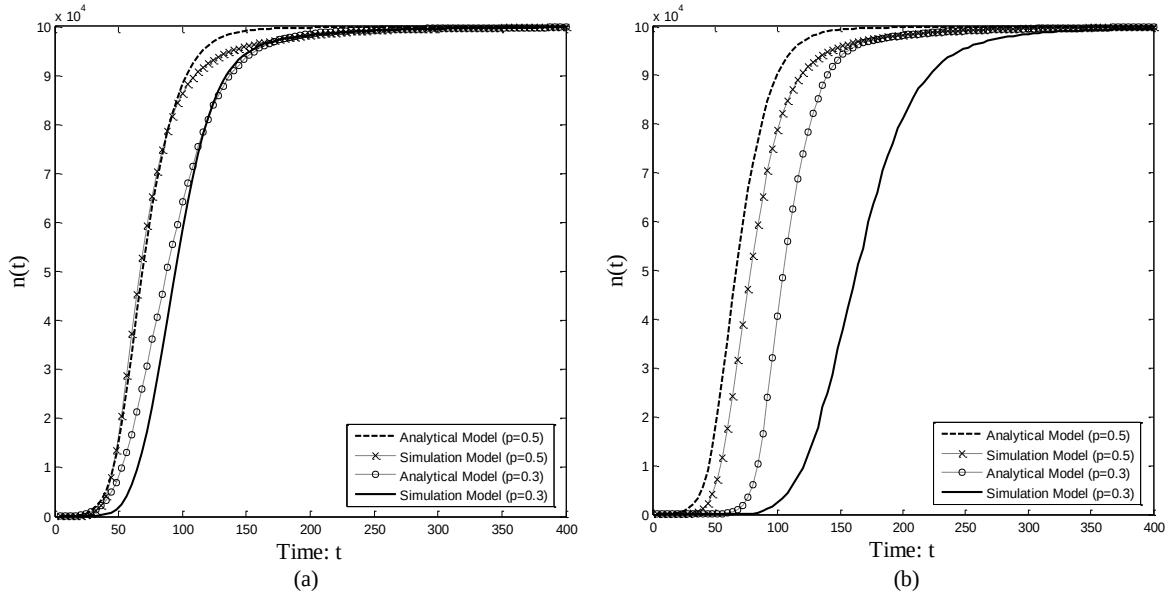



Figure 6.10: The propagation of reinfection worms with different infection probability *p*.
(a) Uncorrelated network with $CT \sim N(40, 20^2)$; (b) Correlated network with $CT \sim N(40, 20^2)$

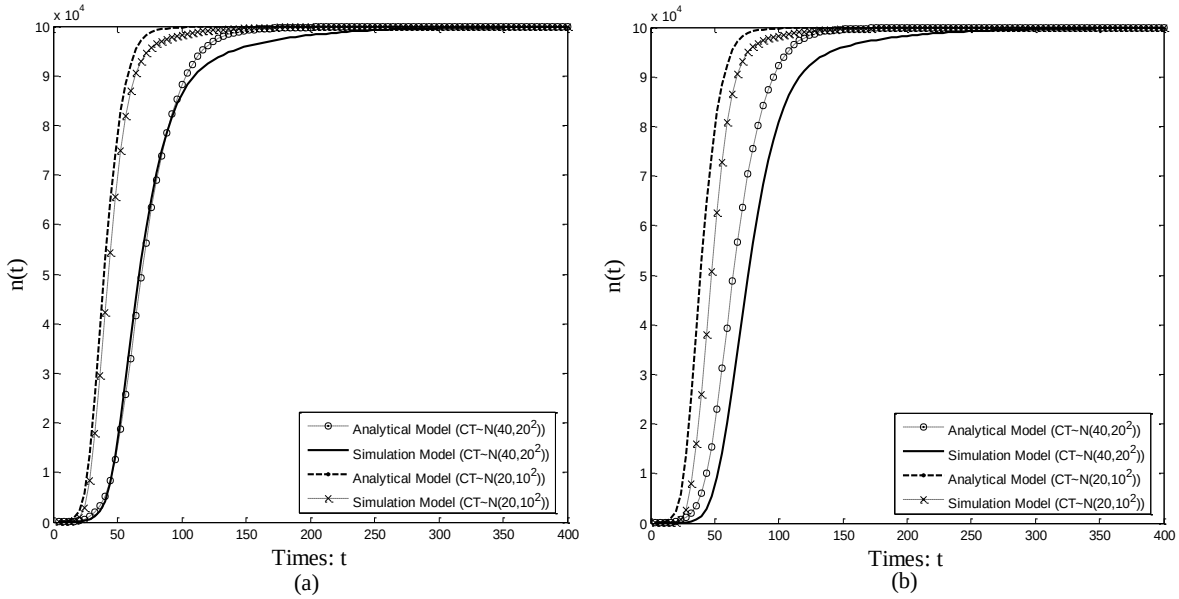



Figure 6.11: The propagation of reinfection worms with different infection probability *p*.
(a) Uncorrelated network with $CT \sim N(40, 20^2)$; (b) Correlated network with $CT \sim N(40, 20^2)$

*Remark* 3: Our analytical model can accurately reflect the propagation of reinfection worms.

The simulation model [6] did not consider the effect of repetitious spreading and assumed only one copy was sent by an infected user, which results in underestimating the scale of infection throughout the network. However, the repetitious spreading can lead to an overwhelming number of emails without considering user awareness. In order to be more

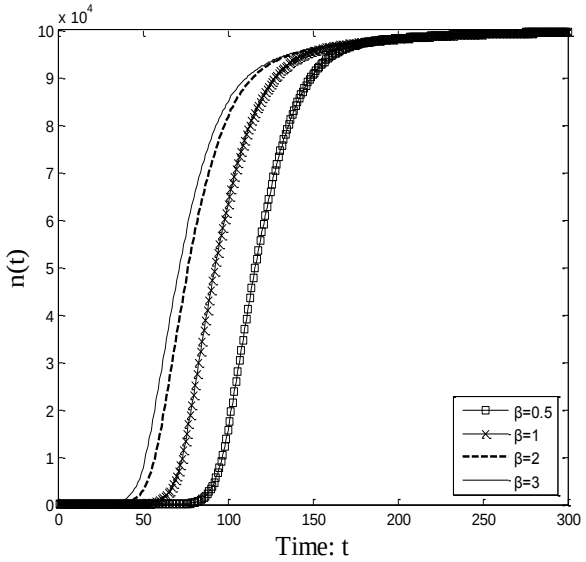



Figure 6.12: Reinfection worms' propagation with $\beta$.

realistic, we introduce virtual users sending malicious copies in the propagation procedure. To overcome the snowball effect, we adopt the vigilance degree $\beta$ to reflect a user's awareness when they receive an abnormal number of emails. As shown in Fig. 6.12, when the value of $\beta$ increases, the spreading speed becomes faster. A high value of $\beta$ represents more malicious copies checked by users so that users can be easier to infect. In our analytical model, this means more virtual users are involved in the propagation procedure. Thus, we have:

*Remark* 4: A large $\beta$ can lead to a more accurate model when the worm is fairly deceptive

However, it is observed that the increase of spreading speed decreases with the increasing $\beta$. This is because the infected probability of virtual users decreases in (6.28) and (6.38). Meanwhile, a large $\beta$ may consume a great deal of computation. Thus, we need to choose a large but suitable $\beta$ for modeling.

## 6.6 Modeling of Self-start Reinfection Worms

### 6.6.1 How Self-start Reinfection Worms Work

Evolving from reinfection, self-start reinfection worms register themselves in start-up services and automatically send out malicious copies once the system starts or specific events are triggered. For example, in Fig. 6.3 (b-3), user 2 sends out a worm email to the victim at time *t*-1 as self-start reinfection. The victim receives five worm emails from their neighbors at time *t*. The state transition graph of an email user is shown as in Fig. 6.1(d). We use $v_s(i,t)$ to indicate the probability of user *i* having been infected at time *t*-1 and being active at time *t* under the scenario of self-start reinfection.

### 6.6.2 The Model

The self-start reinfection propagation procedure is determined by a user's personal habits and, thus, it is independent of the event of $X_i(t\text{-}1)=1$. Different from reinfection, we derive $v_s(i,t)$ as in:

$$\begin{aligned} & v_s(i,t) \\ & = P\left(X_i(t)=1.1 \mid X_i(t-1)=1\right) \\ & = \underbrace{P\left(X_i(t)=1.1 \mid X_i(t-1)=1, start_i(t)=1\right)}_{\text{inevitable event}} P(start_i(t)=1) + \\ & \quad P\left(X_i(t)=1.1 \mid X_i(t-1)=1, start_i(t)=0\right) P(start_i(t)=0) \\ & = Q(i,t) + \underbrace{P\left(X_i(t)=1.1 \mid start_i(t)=0, X_i(t-1)=1\right)}_{f1}\left(1-Q(i,t)\right) \end{aligned} \tag{6.41}$$

According to theorem 3, we have

$$v_s(i,t) = Q(i,t) + v(i,t)\left(1-Q(i,t)\right) \tag{6.42}$$

*Theorem* 3: when the events $X_j(t')=1.1$ and $X_i(t\text{-}1)=1$ are independent, there is $P(X_i(t)=1.1|start_i(t)=1, X_i(t\text{-}1)=1)$ is equal to $v(i,t)$.

*Proof*: similar to (6.15), (6.17) and (6.18), we have the following derivation:

In this work, we assume the events $X_j(t')=1.1$ and $X_i(t\text{-}1)=1$ are independent. Similar to (6.19), we then prove the proposition as in

$$\begin{aligned}
&P\left(X_i(t)=1.1 \middle| start_i(t)=1, X_i(t-1)=1\right) \\
&=\left[1-\prod_{j\in N_i,t'}\left(1-p_{ji}P\left(X_j(t')=1.1\right)\right)\right]G(i,t) \\
&=S(i,t)G(i,t) \\
&=v(i,t)
\end{aligned}$$

Therefore, $f1$ in (6.41) is equal to $v(i,t)$.

In order to model the propagation of self-start reinfection email worms, we extend $f1$ in (6.19) of the basic model. Then we have

$$\begin{aligned}
&s(i,t) \\
&=1-\prod_{j\in N_i,t'}\Big[1-p_{ji}\Big(v(j,t'-1)P\left(X_j(t'-1)=0\right)+ \\
&\quad v_S(j,t'-1)P\left(X_j(t'-1)=1\right)\Big)\Big] \\
&=1-\prod_{j\in N_i,t'}\left[1-p_{ji}\left(\underbrace{v(j,t'-1)}_{f1}+\underbrace{Q(j,t'-1)}_{f2}\left(1-v(j,t'-1)\right)P\left(X_j(t'-1)=1\right)\right)\right]
\end{aligned} \tag{6.43}$$

Similar to (6.34) and (6.35), we can derive (6.44) as in:

$$\begin{aligned}
&s(i,t)=1-(1-G(i,t-1)s(i,t-1)) \\
&\prod_{j\in N_i}\left[1-p_{ji}\left(\underbrace{v(j,t-2)}_{f1}+\underbrace{Q(j,t-2)}_{f2}\left(1-v(j,t-2)\right)P\left(X_j(t-2)=1\right)\right)\right]
\end{aligned} \tag{6.44}$$

As is shown in Fig. 6.9(c), the self-start process can be considered as a virtual user ($N_{i|S}$) who sends worm emails to user $i$ periodically. We disassemble the neighbors of user $i$ into three parts: $N_{i|R}$, $N_{i|V}$ and $N_{i|S}$. For $N_{i|R}$ and $N_{i|V}$, we have $f2=0$ because there is no self-start

effect on them. For $N_{i|S}$, we also have $f1=0$ because this kind of virtual user results from the self-start process rather than from worm spreading. Therefore, we are able to factorize (6.44) as in

$$\begin{aligned} & s(i,t) \\ & = 1 - \underbrace{(1 - G(i,t-1)s(i,t-1)) \prod_{j \in N_{i|R},t'} \left(1 - p_{ji} v(j,t-2)\right)}_{=\lambda} \\ & \quad \prod_{j \in N_{i|V},t'} \left(1 - p_{ji} v(j,t-2)\right) \prod_{j \in N_{i|S},t'} \left(Q(j,t-2) P\left(X_j(t-2)=1\right)\right) \\ & \approx 1 - \lambda \prod_{j \in N_{i|V}(t-2)} \left(1 - p_{ji} P\left(X_{j\_K}(t-2)=1.1\right)\right) \\ & \quad \prod_{j \in N_{i|S},t'} \left( \underbrace{Q(j,t-2) P\left(X_j(t-2)=1\right)}_{f1} \right) \end{aligned} \tag{6.45}$$

$f1$ in (6.44) can be calculated by iteration. Therefore, the number of infected users in the network ($n(t)$) for self-start reinfection worms can be estimated by difference equations (6.2), (6.15) and (6.45).

### 6.6.3 Evaluation of the Self-start Reinfection Worms Model

The difference between the propagation of reinfection and self-start reinfection worms is that the latter can be trigged by specific events and the system restart process. The propagation dynamics of self-start reinfection is similar to the one of reinfection. In this subsection, we mainly analyze the impact of the self-start period *RT* on the propagation procedure. The dependency effect may affect the investigation of the self-start process, and thus, we only examine the propagation of self-start reinfection worms in the uncorrelated network.

In Fig. 6.13, it is observed that the spreading speed is faster if the self-start period is short. If a spreading process can be trigged by more events such as opening a picture or movie files,

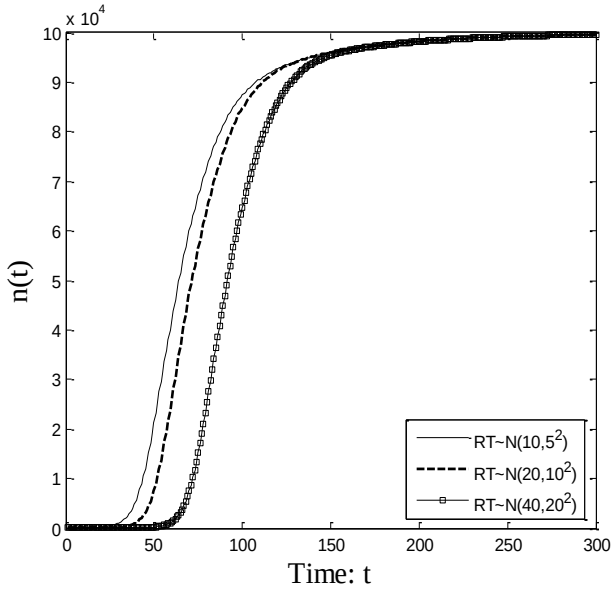



Figure 6.13: The propagation of self-start reinfection worms in an uncorrelated network with *RT*.

it means the worm is more aggressive and has a faster propagation speed. However, in the real world, it is harder for aggressive worms to conceal themselves. Our analytical model can reflect the self-start reinfection propagation process.

### 6.6.4 Comparison of the Spreading Speed of Different Email Worms

We constructed our basic model of worm propagation in Section 6.3.2. The mechanism of spreading varies for different type of worms. As shown in Fig. 6.3, the victim receives three worm emails for non-reinfection, four for reinfection and five for self-start reinfection. In this subsection, we will investigate and compare the spreading speed of each type of worm. We also discuss the reason for their derivation.

*Remark* 5: The spreading speed of reinfection worms is much faster than non-reinfection worms.

We use the number of infected users at time $t(n(t))$ as the benchmark to estimate the spreading speed. The key propagation procedure is described by (6.23) for non-reinfection worms and by (6.37) for reinfection worms. In (6.23), $f2$ belongs to [0, 1] so that $s(i,t)$ of

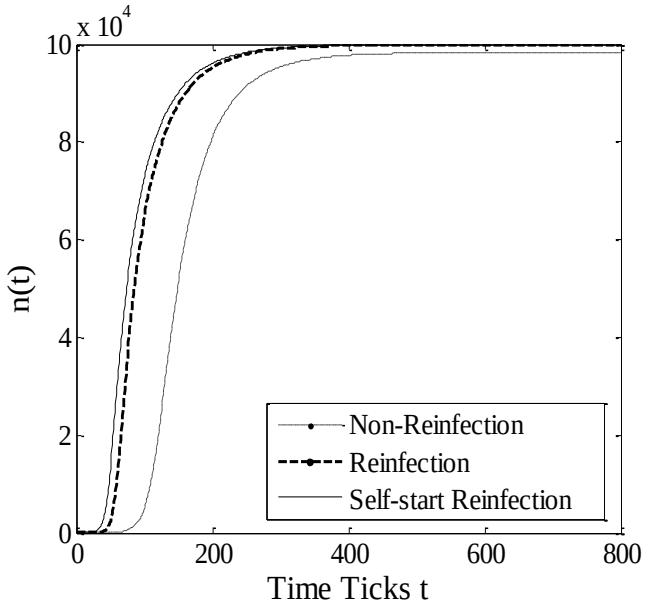



Figure 6.14: The propagation of non-reinfection, reinfection and self-start reinfection worms in an uncorrelated network.

(6.23) is less than (1-$\lambda$) in (6.37). Moreover, (6.37) contains the component which reflects the effect of virtual users in the propagation of worms, and the value of this component is less than one. Thus, it is easy to prove the spreading speed of reinfection worms is much faster than non-reinfection worms.

*Remark* 6: The spreading speed of self-start reinfection worms is much faster than reinfection worms.

The key propagation procedure of self-start reinfection worms is described by (6.43). Compared with reinfection, (6.43) has an extra component $f1$, which reflects the effect of the propagation procedure when the system starts or specific events are triggered. The value of $f1$ is less than one. As a result, the value of (6.37) is less than (6.43). Therefore, the spreading speed of self-start reinfection worms is much faster than reinfection worms.

We compare the spreading speeds and show the difference between different kinds of email worms in Fig. 6.14. We find reinfection and self-start reinfection email worms propagate much faster than non-reinfection email worms.

## 6.7 Summary

This chapter presented a new method for modeling the propagation process of email worms. We discussed three categories of email worms: non-reinfection, reinfection and self-start reinfection. We also analyzed previous research and compared our approach with these works. The evaluation we performed demonstrates the accuracy of our approach. Researchers can employ our analytical model to analyze the propagation of worms in order to provide defense strategies. We believe this is the most significant characteristic and the most important contribution of this thesis.

There is still much work to be done in relation to the propagation of email worms. Firstly, in this chapter, we focused on the modeling propagation procedure for various email worms. As part of our ongoing work, we plan to estimate the parameters of worms' propagation and use our proposed model to study the countermeasures [82] for controlling the spread of email worms. Secondly, by making use of our model, we have studied the impact of the underlying topology on the propagation of worms. However, an email network is essentially a complex network, and many factors of complex networks can affect the propagation and defense of worms but these have not been explored in this work [83-84]. Thirdly, in a correlated email network, the effect on dependence cannot be neglected. In order to analyze the model more accurately, future work will pay attention to eliminating the impact on dependence completely. Readers can find more details in [32] and [80]. Finally, in the real world, infected users may clean the email worms and recover. More comprehensive analysis on the propagation of worms should involve the recovery procedure. In this thesis, we mainly focused on the propagation procedure and thus, our model is based on the *SI* model.

# Chapter 7

# Conclusions and Future Work

This chapter summarizes the main contributions of this thesis on modeling and defenses against worm propagation in networks. It also provides suggestions for improving our research in the future.

## 7.1 Conclusions

In this thesis, we have conducted research on the characterization of worm spreading behavior, analyzed their propagation mechanisms, modeled their propagation procedures and developed defense strategies. The research contributions of this thesis have been made in the following areas.

### 7.1.1 A Microcosmic Model of Worm Propagation

Each year, large amounts of money and labor are spent by the industry on patching vulnerabilities in operating systems and popular software. In order to prevent worms from spreading effectively, many models have been proposed by research and application

communities. Most worm propagation models, however, are based on a macroscopic viewpoint. They focus on the overall tendency of the worm to spread and do not describe the worm propagation from node to node or the infection procedure when disrupted by patching or immunizing nodes. Consequently, the macroscopic model makes it hard to deal with the problems of where, when and how many nodes we need to patch. The question then arises as to how we can develop a model that can accurately reflect the distribution of nodes in the network, which is beneficial for describing the propagation procedure, and thus, can answer the three proposed problems.

In Chapter 3, a microcosmic worm propagation model was proposed. We introduced a complex matrix to represent the probabilities and the time delay between each pair of nodes. These two factors lead to an accurate exploration of the propagation procedure and estimation of both infection scale and the effectiveness of defense. We also developed three vectors for investigating the different scenarios of infectious states, vulnerable states and quarantine states. Compared with a macrocosmic propagation model, a microcosmic model prefers to study the dynamic propagation between nodes and is able to understand how the current infected states impact on the worm's propagation in the next step. In addition, we introduced an error calibration vector for analyzing the errors caused by reinfection in macroscopic models. Modeling a microcosmic propagation procedure can provide defenders with useful information to deal with the problems of where, when and how many nodes we need to patch.

### 7.1.2 Defense Study against Scanning Worms

Scanning is one of the most common strategies employed by worms for spreading. Scan-based worms (scanning worms) probe the entire network and infect targets without regard to topological constraints. It is closely related to the logical features of the network rather than

the physical structure. The objective of studying scanning worms is to address the three practical aspects of preventing worm propagation: where, when and how many nodes we need to patch.

In Chapter 4, we used Code Red II as an example to evaluate the vulnerability distribution and patch strategy vector from the microcosmic worm propagation model in Chapter 3 and presented a series of recommendations and advice for immunization defense. Firstly, the IP ranges with a high density of vulnerable nodes are essential areas for patching. Secondly, for high risk vulnerabilities, it is critical that networks reduce the number of vulnerable nodes to below a certain threshold, e.g., 80% in this analysis. Thirdly, increased disclosure of specific vulnerabilities could possibly be delayed until the patching rate reaches a certain threshold, e.g., at least 20% in this analysis. Moreover, we observed the effect of different impact factors step by step, which reflected the mutual impact between the propagation probability and time delay. Experimental results indicated that an increase in time delay results in a small propagation probability of the worm's propagation. In addition, the overestimation in macroscopic models caused by propagation cycles was also discussed.

### 7.1.3 Defense Study against Topology-based Worms

Topology information is a fundamental element that enables topology-based worms, such as email worms and social network worms. In order to control the impact of their outbreak, large amounts of money and labor are spent on devising effective strategies for defense. Questions then arise as to how to model the propagation mechanism of topology-based worms so that we can provide effective schemes to deal with the problems of where and how many nodes we need to patch to prevent them from propagating.

In Chapter 5, a novel probability matrix was proposed to model the spreading of topology-based worms. We introduced a propagation source vector and a patch strategy vector to evaluate their effects in different spreading scenarios and investigate a more effective immunization defense for preventing worms from propagating. We take a typical topology-based worm, such as an email worm, as an example investigating how a worm spreads from one node to another node through a group of intermediate nodes. Through model analysis, we derive a better understanding of dynamic infection procedures in each step to answer the proposed questions. The results from experiments showed that, for a power law network, a more effective patch strategy against email worm propagation is to immunize the most-connected nodes. Besides this, the effect of random patching is not obvious as email worm spreading relies on the underlying connectivity between each pair of nodes. In addition, we analyzed the formation of propagation errors and examined the impact of eliminating errors on the propagation procedure of topology-based worms. We have shown through simulations that errors increase as more propagation cycles are formed and quantified the errors under different propagation scenarios. This work is helpful for the accurate analysis of worm spreading.

### 7.1.4 Modeling the Propagation Dynamics of Email Worms

Spreading malicious code through email is still effective and is widely used by current attackers. However, previous work has preferred to rely on simulation modeling rather than on mathematical analysis because of the following two aspects. Firstly, each user has their own habits for checking emails. It is really hard to characterize the propagation dynamics with different mailbox checking time between email users in a large scale network. Secondly,

modern email worms belong to reinfection or self-start reinfection worms. This means it is difficult to model the repetitious email sending process.

In Chapter 7, an analytical model was proposed to characterize the propagation dynamics of email worms. Our model extensively investigates different classes of real-world worms based on their infection strategies, including non-reinfection, reinfection, and modern self-start reinfection categories. We examined the individual steps and state transitions in the propagation procedure. Compared to the simulation model [6] and the spatial-temporal model [32], our model can provide an accurate representation of the propagation of worms with different checking time of mailboxes from users. We also analyzed the propagation mechanisms of reinfection and self-start reinfection worms respectively. In particular, the concept of virtual users was introduced to represent the process of sending repetitive emails. Therefore, our model can accurately reflect the propagation of reinfection and self-start reinfection worms. The results from our experiments indicate that our analytical model is accurate and helpful in providing a better and more realistic understanding of the propagation of email worms.

## 7.2 Future Work

There are a number of areas where future work can be pursued.

- **Characterizing the propagation of social network worms**: The spreading of social network worms rely on the topology of social networks, which may result in a problem of spatial dependence in the propagation procedure. This means that compromised users will infect their neighbors but the probabilities for those compromised users being infected may be due to their neighbors having been infected before and then spreading the worm to these compromised users. This

results in redundant computation of infection probabilities. In order to simplify this problem, some research has assumed the status of all nodes at each time tick to be spatially independent. However, it is a weak approximation to the spreading dynamics. Therefore, we will attempt to discover what the spatial dependence is and how we can approximate it so that we can eliminate the redundancy and describe the real spreading probability.

- **Locating defense positions**: The centrality of a node in a social network is a measure of its structural importance and prominence in the group. It can be calculated in a number of ways depending on whether one measures it in terms of the degree, the closeness or the betweenness. In this thesis, we have proved that patching the highly-connected (high degree) nodes is an effective immunization defense for preventing topology-based worms from spreading. Under certain conditions, however, such as when some popular users (highly-connected nodes in the network) have more vigilance of malicious codes, this may not always be the truth. Therefore, how to locate more suitable positions through a measure of betweenness and closeness for slowing down the worm propagation should be considered in future research.

- **Overhead analysis**: The proposed model in the thesis investigates the propagation probability between each pair of nodes through matrix computation. If the simulated network has a larger scale, the simulation overhead can be prohibitively high in some cases. In the real world, however, some nodes have no direct connection in the Internet. If those nodes can be removed from the proposed model, the simulation overhead can be saved to a certain extent. Therefore, how to represent the nodes without direct connection in the Internet in the proposed model should be considered in future work.

- **Attack source analysis**: In the real world, the distribution and number of attack sources has considerable impact on the spreading speed of worms. Understanding the topology of the entire network has been a great help for defenders in analyzing the attack sources and then deploying the immunization defense. However, except for ISPs and administrators of social networks, it is hard to obtain the structure of the entire network. The problem then arises as to how we can minimize the number of possible attack sources with only partial network knowledge and effectively prevent worm propagation.

# Bibliography

[1] Y. Xiang, X. Fan, W. T. Zhu, Propagation of active worms: A survey, Computer Systems Science & Engineering 24 (2009) 157-172.

[2] C. C. Zou, W. Gong, D. Towsley, Code red worm propagation modeling and analysis, in: Proceedings of the 9th ACM conference on Computer and communications security, in CCS '02, ACM, New York, NY, USA, 2002, pp. 138-147.

[3] C. C. Zou, L. Gao, W. Gong, D. Towsley, Monitoring and early warning for internet worms, in: Proceedings of the 10th ACM conference on Computer and communications security, in CCS '03, ACM, New York, NY, USA, 2003, pp. 190-199.

[4] C. C. Zou, D. Towsley, W. Gong, S. Cai, Routing worm: A fast, selective attack worm based on ip address information, in: Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation, PADS '05, IEEE Computer Society, Washington, DC, USA, 2005, pp.199-206.

[5] C. C. Zou, D. Towsley, W. Gong, On the performance of internet worm scanning strategies, Elsevier Journal of Performance Evaluation 63 (2003) 700-723.

[6] C. C. Zou, D. Towsley, W. Gong, Modeling and simulation study of the propagation and defense of internet e-mail worms, Dependable and Secure Computing, IEEE Transactions on 4 (2007) 105-118.

[7] S. Staniford, V. Paxson, N. Weaver, How to own the internet in your spare time, in: Proceedings of the 11th USENIX Security Symposium, USENIX Association, Berkeley, CA, USA, 2002, pp. 149-167.

[8] K. Rohloff, T. Basar, Stochastic behavior of random constant scanning worms, in: Proceedings of the 14th International Conference on Computer Communications and Networks, ICCCN '05, San Diego, CA, USA, 17-19 October, 2005, pp. 339-344.

[9] S. Sellke, Modeling and automated containment of worms, in: Proceedings of the 2005 International Conference on Dependable Systems and Networks, DSN '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 528-537.

[10] Z. Chen, L. Gao, K. Kwiat, Modeling the spread of active worms, in: Proceeding of The 22th Annual Joint Conference of the IEEE Computer and Communications, INFOCOM'03, IEEE Societies, San Franciso, CA, USA, 30 March-3 April, 2003, pp. 1890-1900.

[11] J. Wu, S. Vangala, L. Gao, An efficient architecture and algorithm for detecting worms with various scan techniques, in: Proceedings of the 11th Annual Network and Distributed System Security Symposium ,NDSS'04, The Internet Society, San Diego, CA, USA, 2004, pp. 143-156.

[12] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver, Inside the slammer worm, IEEE Security and Privacy 1 (2003) 33-39.

[13] J. Daley, J. Gani, Epidemic modeling: An introduction, Cambridge University Press, Cambridge (1999).

[14] N. Weaver, V. Paxson, S. Staniford, R. Cunningham, A taxonomy of computer worms, in: Proceedings of the 2003 ACM workshop on Rapid malcode, WORM '03, ACM, New York, NY, USA, 2003, pp. 11-18.

[15] D. Moore, C. Shannon, K. Claffy, Code-red: a case study on the spread and victims of an internet worm, in: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, IMW '02, ACM, New York, NY, USA, 2002, pp. 273-284.

[16] eEye Digital Security, Advisories and Alerts: .ida "Code Red" Worm, July 2001. Available: http://www.eeye.com/Resources/Security-Center/Research/Security-Advisories/AL20010717.

[17] eEye Digital Security, blaster worm analysis, July 2001.
Available: http://www.eeye.com/Resources/Security-Center/Research/Security-Advisories/ AL20030811.

[18] R. Thommes, M. Coates, Epidemiological modeling of peer-to-peer viruses and pollution, in: Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM '06, pp. 1-12.

[19] X. Fan, Y. Xiang, Modeling the propagation of peer-to-peer worms, Future Gener. Comput. Syst. 26 (2010) 1433-1443.

[20] W. Fan, K. Yeung, Online social networks paradise of computer viruses, Physica A: Statistical Mechanics and its Applications 390 (2011) 189-197.

[21] G. Yan, S. Eidenbenz, Modeling propagation dynamics of bluetooth worms (extended version), Mobile Computing, IEEE Transactions on 8 (2009) 353-368.

[22] G. Yan, G. Chen, S. Eidenbenz, N. Li, Malware propagation in online social networks: nature, dynamics, and defense implications, in: Proceedings of the 6th

ACM Symposium on Information, Computer and Communications Security, ASIACCS '11, ACM, New York, NY, USA, 2011, pp. 196-206.
[23] R. Anderson, R. May, Infectious Diseases of Humans: Dynamics and Control, Oxford University Press, Oxford, 1991.

[24] H. Andersson, T. Britton, Stochastic Epidemic Models and their Statistical Analysis, Springer, NewYork, 2000.

[25] T. Bailey, The Mathematical Theory of Infectious Diseases and its Application, Hafner Press, NewYork, 1975.

[26] C. Frauenthal, Mathematical Models in Epidemiology, Springer, NewYork, 1980.

[27]W32.Koobface,http://www.symantec.com/security_response/writeup.jsp?docid=2008-080315-0217-99.

[28] N. Weaver, A brief history of the worm, 2001.
http://www.symantec.com/connect/articles/brief-history-worm.

[29] CAIDA. CAIDA analysis of code-red. (12 August 2011, date online accessed).
http://www.caida.org/analysis/security/code-red/.

[30] D. Moore, The spread of the code-red worm (crv2), 2001.
http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml.

[31] G. Kesidis, I. Hamadeh, S. Jiwasurat, Coupled kermack-mckendrick models for randomly scanning and bandwidth-saturating internet worms, in: Proceedings of the Third international conference on Quality of Service in Multiservice IP Networks, QoS-IP'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 101-109.

[32] Z. Chen, C. Ji, Spatial-temporal modeling of malware propagation in networks, Neural Networks, IEEE Transactions on 16 (2005) 1291-1303.

[33] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, B. Bhattacharjee, Measurement and analysis of online social networks, in: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, IMC '07, ACM, New York, NY, USA, 2007, pp. 29-42.

[34] M. Boguna, R. Pastor-Satorras, and A. Vespignani, Epidemic spreading in complex networks with degree correlations, in Statist. Mech. Complex Netw., R. Pastor-Satorras et al., Eds., 2003, pp. 127–147.

[35] P. Erdos, A. Renyi, On the evolution of random graphs, in: Publication of the Mathematical, Institute of the Hungrian Academy of Sciences, vol. 5, 1960, pp. 17–61.

[36] Y. Wang, S. Wen, S. Cesare, W. Zhou, Y. Xiang, The microcosmic model of worm propagation, Comput. J. 54 (2011) 1700-1720.

[37] J. Kleinberg, R. Rubinfeld, Short paths in expander graphs, in: Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS '96, IEEE Computer Society, Washington, DC, USA, 1996, pp. 86-95.

[38] M. Molloy, B. Reed, A critical point for random graphs with a given degree sequence, Random Structures and Algorithms 6 (1995) 161-179.

[39] M. Molloy, B. Reed, The size of the giant component of a random graph with a given degree sequence, COMBIN. PROBAB. COMPUT 7 (2000) 295-305.

[40] M. Jovanovic, F. Annexstein, K. Berman, Modeling peer-to-peer network topologies through small-world models and power laws, in: Telecommunications Forum, Nov. 2001.

[41] C. Shannon, D. Moore, The spread of the witty worm, IEEE Security and Privacy 2 (2004) 46-50.

[42] Symantec, W32.Sasser.Worm, 2004,
Available: http://www.symantec.com/security_response/writeup.jsp?docid=2004-050116-1831-99

[43] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the web for emerging cyber-communities, Comput. Netw. 31 (1999) 1481-1493.

[44] A. L. Barabasi, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509-512.

[45] V. Braitenberg and A. Schuz, Anatomy of a Cortex: Statistics and Geometry. Springer-Verlag, Berlin, 1991.

[46] L. A. Adamic, O. Buyukkokten, E. Adar, A Social Network Caught in the Web, First Monday 8(6) (2003).

[47] H. Ebel, L. Mielsch, and S. Bornholdt, Scale-free topology of e-mail networks, Phys. Rev. E 66, 035 103(R) (2002).

[48] N. Li, G. Chen, Analysis of a location-based social network, in: Proceedings of the 2009 International Conference on Computational Science and Engineering, Volume 04, CSE '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 263-270.

[49] R. Kumar, J. Novak, A. Tomkins, Structure and evolution of online social networks, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06, ACM, New York, NY, USA, 2006, pp. 611-617.

[50] Y. Y. Ahn, S. Han, H. Kwak, S. Moon, H. Jeong, Analysis of topological characteristics of huge online social networking services, in: Proceedings of the 16th international conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 835-844.
[51] D. J. Watts, S. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998) 440-442.

[52] X. Song, B.P. Paris, Measuring the size of the internet via importance sampling, IEEE Journal on selected areas in communications 21 (2003) 922- 933.

[53] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks, Phys. Rev. Lett. 86 (2001) 3200-3203.

[54] Y. Moreno, J. B. Gomez, A. F. Pacheco, Epidemic incidence in correlated complex networks, Phys. Rev. E 68 (2003) 035103.

[55] Y. Moreno, R. Pastor-Satorras, A. Vespignani, Epidemic outbreaks in complex heterogeneous networks, Eur. Phys. J. B 26 (2002) 521-529.

[56] C. Wang, J. Knight, M. Elder, On computer viral infection and the effect of immunization, in: Computer Security Applications, 2000. ACSAC'00. 16th Annual Conference, pp. 246-256.

[57] A. Vazquez, B. Racz, A. Lukacs, A.L. Barabasi, Impact of nonpoissonian activity patterns on spreading processes, Phys. Rev. Lett.98 (2007) 158702.

[58] Symantec, W32.imsolk.b@mm technical details, 2011.
Available: http://www.symantec.com/security_response/writeup.jsp?docid=2010-090922-4703-99&tabid=2

[59] M. Fossi, J. Blackbird, Symantec Internet Security Threat Report 2010, Technical Report, Symantec Corporation, March, 2011.

[60] J. Kim, S. Radhakrishnan, S. Dhall, Measurement and analysis of worm propagation on internet network topology, in: Proceedings 13th International Conference on Computer Communications and Networks, 2004. ICCCN 2004, pp. 495-500.

[61] Y. Wang, C. Wang, Modeling the effects of timing parameters on virus propagation, in: Proceedings of the 2003 ACM workshop on Rapid malcode, WORM '03, ACM, New York, NY, USA, 2003, pp. 61-66.

[62] M. E. J. Newman, S. Forrest, J. Balthrop, Email networks and the spread of computer viruses, Phys. Rev. E 66 (2002) 035101.

[63] S. M. Cheng, W. C. Ao, P. Y. Che, K. C. Chen, On modeling malware propagation in generalized social networks, IEEE Communications Letters, 15 (2011) 25-27.

[64] Symantec Global Internet Security Threat Report. http://eval.symantec.com/mktginfo/enterprise/white_papers/bwhitepaper_internet_security_threat_report_xv_04-2010. enus.pdf. (12 August 2011, date online accessed).

[65] S.J. Horng, M.Y. Su, Y.H. Chen, T.W. Kao, R.J. Chen, J.L. Lai, C. D. Perkasa, A novel intrusion detection system based on hierarchical clustering and support vector machines, Expert Syst. Appl. 38 (2011) 306-313.

[66] M. Bailey, E. Cooke, F. Jahanian, D. Watson, J. Nazario, The blaster worm: Then and now, IEEE Security Privacy, 3 (2005) 26-31.

[67] T. Forbath, P. Kalaher, and T. O'Grady, The Total Cost of Security Patch Management, Wipro Report, Wipro Technologies, Bangalore, India (2005).

[68] D. Schneider, Fresh phish. IEEE Spectr., 45 (2008) 34–38.

[69] CAIDA. The Spread of the Sapphire/Slammer Worm.
Available: http://www.caida.org/publications/papers/2003/sapphire/sapphire.html.
(12 August 2011, date online accessed).

[70] Wikipedia, Melissa computer virus, 2011.
Available: http://en.wikipedia.org/wiki/Melissa_(computer_virus).

[71] F-Secure, Love letter virus, 2011.
Available: http://www.f-secure.com/v-descs/love.shtml.

[72] Symantec, W32.mydoom.a@mm technical details, 2011.
Available: http://www.symantec.com/security_response/writeup.jsp?docid=2004-012612-5422-99&tabid=2.

[73] B. Rozenberg, E. Gudes, Y. Elovici, Sisr -- a new model for epidemic spreading of electronic threats, in: Proceedings of the 12th International Conference on Information Security, ISC '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 242-249.

[74] M. C. Calzarossa, E. Gelenbe, Performance Tools and Applications to Networked Systems: Revised Tutorial Lectures (Lecture Notes in Computer Science), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004.

[75] D. Chakrabarti, J. Leskovec, C. Faloutsos, S. Madden, C. Guestrin, M. Faloutsos, Information survival threshold in sensor and p2p networks, in: INFOCOM 2007. The 26th IEEE International Conference on Computer Communications, IEEE, pp. 1316-1324.

[76] A. Ganesh, L. Massoulie, D. Towsley, The effect of network topology on the spread of epidemics, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2005, vol. 2, pp. 1455-1466.

[77] Y. Wang, D. Chakrabarti, C. Wang, C. Faloutsos, Epidemic spreading in real networks: an eigenvalue viewpoint, in: Proceedings of the 22nd International Symposium on Reliable Distributed Systems, 2003, pp. 25-34.

[78] P. Manna, S. Chen, S. Ranka, Inside the permutation-scanning worms: Propagation modeling and analysis, IEEE/ACM Transactions on Networking, 18 (2010) 858-870.

[79] C. C. Zou, W. Gong, D. Towsley, L. Gao, The monitoring and early detection of internet worms, IEEE/ACM Transactions on Networking, 13 (2005) 961-974.

[80] Y. Wang, S. Wen, S. Cesare, W. Zhou, Y. Xiang, Eliminating errors in worm propagation models, IEEE Communications Letters, 15 (2011) 1022-1024.

[81] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the internet topology, SIGCOMM Comput. Commun. Rev. 29 (1999) 251-262.

[82] Y. Y. Liu, J. J. Slotine, B. Hungary, A. lszl Barabsi, Controllability of complex networks, Nature 473 (2011) 167-173.

[83] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814-818.

[84] G. Palla, A. lszl Barabsi, T. Vicsek, B. Hungary, Quantifying social group evolution, Nature 446 (2007) 664-667.

[85] Symantec. Symbos.commwarrior worm description. http://securityresponse.symantec.com/avcenter/venc/data/symbos.commwarrior.a.html, October 2005.

[86] H. Berghel, The code red worm: Malicious Software Knows No Bounds, Commun. ACM 44 (2001) 15-19.

[87] A.L. Foster, Colleges Brace for the Next Worm, Chronicle of Higher Education, vol. 50, no. 28, 2004, p. A29.

[88] V. Weafer, Downadup/Conficker and April Fool's Day: One Year Later, 2010. Available: http://www.symantec.com/connect/blogs/downadupconficker-and-april-fool-s-day-one-year-later

[89] E. Levy, Worm propagation and generic attacks, IEEE Security and Privacy 3 (2005) 63-65.

[90] ILOVEYOU Virus Lessons Learned Report, Army Forces Command, Available:http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA415104&Location=U2&doc=GetTRDoc.pdf

[91] Symantec, W32.Sircam.Worm, 2001, Available: http://www.symantec.com/security_response/writeup.jsp?docid=2001-071720-1640-99&tabid=2

[92] Symantec, W32.Downadup (Win32/Conficker), 2008, Available: http://www.symantec.com/security_response/writeup.jsp?docid=2008-112203-2408-99

[93] X. Fan, Y. Xiang, Defending against the propagation of active worms, The Journal of Supercomputing 51 (2010) 167-200.

[94] Y. Tang, J.Q Luo, B. Xiao and G.Y. Wei, Concept, Characteristics and Defending Mechanism of Worms (invited paper), IEICE Transactions on Information and Systems, 5 (2009) pp. 799-809.

[95] Z. Chen, C. Ji, Importance-scanning worm using vulnerable-host distribution, in: Global Telecommunications Conference, 2005. GLOBECOM '05, IEEE, volume 3, pp. 1779-1784.

[96] Z. Chen, C. Ji, Optimal worm-scanning scanning method using vulnerable-host distributions, Int. J. Secur. Netw. 2 (2007) 71-80.

[97] Z. Chen, C. Chen, A closed-form expression for static worm-scanning strategies, in: IEEE International Conference on Communications, 2008. ICC '08, pp. 1573-1577.

[98] Z. Chen, C. Chen, C. Ji, Understanding localized-scanning worms, in: IEEE International Performance, Computing, and Communications Conference, 2007. IPCCC 2007, pp. 186-193.

[99] Q. Wang, Z. Chen, C. Chen, N. Pissinou, On the robustness of the botnet topology formed by worm infection, in: IEEE Global Telecommunications Conference (GLOBECOM 2010), 2010, pp. 1-6.

[100] W. Yu, X. Wang, P. Calyam, D. Xuan, W. Zhao, On detecting camouflaging worm, in: the 22nd Annual Computer Security Applications Conference, 2006. ACSAC '06, pp. 235-244.

[101] W. Yu, X. Wang, P. Calyam, D. Xuan, W. Zhao, Modeling and detection of Camouflaging worm, IEEE Transactions on Dependable and Secure Computing, 8 (2011) 377-390.

[102] L. A. Adamic, O. Buyukkokten, E. Adar, A social network caught in the Web, First Monday, 8(6), 2003.