2005

Graph-theoretic Approach To Modeling Propagation And Control Of Network Worms

Zoran Nikoloski
University of Central Florida

GRAPH-THEORETIC APPROACH TO MODELING PROPAGATION AND CONTROL OF NETWORK WORMS

by

ZORAN NIKOLOSKI
B.S. Graceland University, 2001

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2005

ABSTRACT

In today's network-dependent society, cyber attacks with network worms have become the predominant threat to confidentiality, integrity, and availability of network computing resources. Despite ongoing research efforts, there is still no comprehensive network-security solution aimed at controling large-scale worm propagation.

The aim of this work is fivefold: (1) Developing an accurate combinatorial model of worm propagation that can facilitate the analysis of worm control strategies, (2) Building an accurate epidemiological model for the propagation of a worm employing local strategies, (3) Devising distributed architecture and algorithms for detection of worm scanning activities, (4) Designing effective control strategies against the worm, and (5) Simulation of the developed models and strategies on large, scale-free graphs representing real-world communication networks.

The proposed *pair-approximation model* uses the information about the network structure—order, size, degree distribution, and transitivity. The empirical study of propagation on large scale-free graphs is in agreement with the theoretical analysis of the proposed pair-approximation model. We, then, describe a natural generalization of the classical cops-and-robbers game—a combinatorial model of worm propagation

and control. With the help of this game on graphs, we show that the problem of containing the worm is NP-hard. Six novel near-optimal control strategies are devised: combination of static and dynamic immunization, reactive dynamic and invariant dynamic immunization, soft quarantining, predictive traffic-blocking, and contact-tracing. The analysis of the predictive dynamic traffic-blocking, employing only local information, shows that the worm can be contained so that 40% of the network nodes are not affected. Finally, we develop the *Detection via Distributed Blackholes* architecture and algorithm which reflect the propagation strategy used by the worm and the salient properties of the network. Our distributed detection algorithm can detect the worm scanning activity when only 1.5% of the network has been affected by the propagation. The proposed models and algorithms are analyzed with an individual-based simulation of worm propagation on realistic scale-free topologies.

To my parents.

ACKNOWLEDGMENTS

I thank my advisor, Professor Narsingh Deo, for his guidance and assistance, which significantly contributed to the improvement of my research and writing skills. His confidence and unreserved support are greatly appreciated. I would also like to thank the committee members, professors Mostafa Bassiouni, Ronald Dutton, and Yue Zhao, whose comments helped me improve this dissertation. I am also thankful to Professor Ludek Kučera for his assistance during my research visit at Charles University, Prague.

I am grateful to my friends and family for their patience and *love*. Their unconditional support and encouragement are invaluable.

I wish to thank my colleagues for their comments throughout the course of my research. I am also grateful to the entire Computer Science office staff, who were always helpful.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In today's network-dependent society, cyber attacks have become the predominant threat to confidentiality, integrity, and availability of network computing resources. Due to the convergence of features, such as Internet's globally-distributed nature, mass-production of personal computers, and rapid software development, malicious activities remain an ever-increasing problem. Recent cyber attacks have inflicted considerable damage in terms of unsolicited consumption of network bandwidth, degraded corporate productivity (as a result of nonfunctional networks with thousands of computers), and compromised integrity of valuable data. The existing local, automated measures such as anti-virus software, firewalls, and intrusion detection systems, although essential, do not provide complete protection from cyber attacks. As the vital sectors of the *critical infrastructure*—defense, energy, telecommunications, transportation, finance, emergency services, and governmental services—have been integrated via the *Internet*, the disruption of its functionality can have serious political, financial, and tactical consequences.

Cyber attacks employ *malicious mobile-code* (*MMC*)—a program designed to propagate copies of itself to computers on a network by exploiting some *vulnerability* (*i.e.*, weakness in design or implementation of software) to perform a malicious action. One can classify MMCs into three broad categories based on the extent to which human intervention is required for their propagation [Nac99], namely: autonomous, human-dependent, and hybrid. Furthermore, MMCs may be grouped into three classes based on their actions: Trojan horses, computer viruses, and network worms. A *Trojan horse* is a program that is given (fraudulently) the same name as a legitimate piece of software, but, when executed, it performs a malicious act. A *computer virus* is an MMC that modifies resident programs to perform malicious actions. Like a Trojan horse, a computer virus requires human intervention to propagate on a network.

A *network worm* is a stand-alone program that propagates autonomously by sending copies of itself to other computers on the network. Despite increasing efforts and expenditure on cyber-security, the problem of network worms is worsening every year [Ins04]. Due to the short time required for their propagation, in the range from a few milliseconds (for a local host) to a second (for a remote host) [Wea02], worms can inflict considerable damage to networks. For instance, the versions of CodeRed Worm infected 150,000 computer systems in 14 hours [Cen01a]. The damage inflicted by Nimda [Cen01b] to 86,000 computer systems, has been estimated to $13

billion [Ins04]. The Sapphire Worm (SQL Slammer) spread across different networks around the world overnight [MPS03].

Recent studies [LN04] have found the topology on which the worm propagates to be significantly different from the underlining network infrastructure. This is particularly true for network worms that scan the Internet IP space uniformly at random, and therefore can communicate with any computer on the network. Although the introduction of the Internet has arguably made the assumption of sparseness (of the propagation topology) no longer valid, the idea of *locality*, especially in the case of analytical modeling of local propagation strategies [Wea02] (seen with worms such as Code Red II and Nimda) is still applicable. Moreover, as argued by some authors [WPS03b], [SGJ01], for a worm to propagate in the larger IPv6 address space, it will have to use local propagation strategies.

1.1 Problem Statement

The global impact that worms have on today's network-dependent society demands understanding not only of the worm propagation dynamics but also of the feasibility for worm detection and control. Advances in distributed and mobile sensor technologies have made it possible to deploy sensor networks and mobile agents in different environments. To make use of these emerging technologies in network security, it is

crucial to design algorithms that take input from sensors and provide coordination among autonomous agents that control the malicious propagation on a network.

This thesis addresses five problems: (1) Developing an accurate combinatorial model of worm propagation that can facilitate the analysis of worm quarantining. We use the term *quarantining* to denote containment of the worm (*i.e.*, all copies of the MMC) in a set of computers from where propagation to the rest of the network is impossible due to the presence of agents performing security measures. We assume that every computer on the network has the vulnerability exploited by the worm, and that the agents are informed of the worm's presence and propagation strategy by some overlay network of intrusion-detection sensors. Our model of propagation and quarantining is a natural generalization of the classical cops-and-robbers game. We use a variant of the generalized cops-and-robbers game to study the complexity of controlling a worm by quarantining. (2) Building an accurate epidemiological model for the propagation of a worm and the damage inflicted on the network. Our *pair-approximation model* uses the information about the graph structure—order, size, degree distribution, and clustering coefficient (transitivity). Modeling the propagation of network worms is used here to gain quantitative understanding of the vulnerabilities of large real-world networks—expressed in terms of the number of infectious and removed (disabled) nodes. (3) Devising distributed architecture and algorithms for detection of worm scanning activities. Our *Detection via Distributed Blackholes*

($DDBH$) architecture and algorithm reflects both, the propagation strategy used by the attacking worm and the salient properties of the underlying scale-free network, and, in turn, assists in predicting the course of a worm, so as to allocate resources needed by the control strategies. (4) Designing effective control strategies against the propagating worm. Our control strategies—predictive dynamic traffic-blocking and contact-tracing—can be used by the members (agents) of the DDBH architecture to plan and coordinate the activities of the autonomous agents performing security measures; and (5) Simulation of the developed models and strategies on large, scale-free graphs representing real–world communication networks. Our individual-based simulation of worm propagation and control strategies on scale-free graphs is a first attempt to simulate worm's propagation on realistic topologies.

Organization of the Thesis: In Chapter 2, an adequate model of the propagation medium—a scale-free random graph—is described. Two models of the Internet are defined, Macroscopic Internet graph and Microscopic Internet graph, which are later used for simulation purposes. Chapter 3 presents a prototype of a worm and formalization of existing propagation strategies. In Chapter 4, we define a natural generalization of the cops-and-robbers game, and study a cops-and-robbers model of worm propagation and quarantining. In Chapter 5, after presenting a classification of existing models of propagation, we describe the pair-approximation epidemiological model that uses information about the network structure along with simulation

results. Chapter 6 provides a classification of existing control strategies and a description of four novel, worm control strategies. Finally, in Chapter 7 the distributed detection mechanism—Detection via Distributed Blackholes—is presented and coupled with a control strategy called contact-tracing.

CHAPTER 2

MODELS OF REAL-WORLD NETWORKS

2.1 Introduction

A worm propagates on a network by traversing the communication links. Due to the diversity of exploited vulnerabilities, network worms can propagate on a physical or a logical (virtual) network. For instance, the Internet is a representative of a physical network, while the World Wide Web (WWW) and an e-mail network are virtual networks build on top of a physical network. In either case the network is modeled as a graph $G = (V, E)$. Recent empirical studies of the Internet, the WWW, and various e-mail networks have shown statistical similarities between these and other networks, as diverse as the network of phone-calls, power network, citation (also called scientific collaboration) network, movie-actor collaboration network, and the network of sexual contacts. The extent to which these, so-called, *scale-free networks* pervade, influence, and condition our network-dependent society, have prompted the study of *scale-free random graphs*. Analyses of these random graphs can be employed to

design or transform a network in a way that a given purpose, such as effective control strategies against a worm, could be reached in an efficient way.

The three salient properties of the *scale-free networks* are: (1) scale-free degree distribution, (2) large clustering coefficient, and (3) small average distance. Let $V(G)$ be the set of nodes, $E(G)$ the set of edges, n denote the number of nodes and m, the number of edges of graph G.

Degree distribution: The *degree distribution* gives the probability that a node u, chosen uniformly at random from $V(G)$, is of degree d. Empirical studies of real-world networks have demonstrated that the degree distribution falls in the class of so-called *scale-free (power-law) probability distributions*, such that $P(d(u) = d) \sim d^{-f}$ [Mit04]. Here, f is the *exponent* of the scale-free degree distribution.

Clustering coefficient and transitivity: Given a graph $G = (V, E)$ and a node $u \in V$ of degree $d(u)$, the *clustering coefficient* C_u *of node* u is defined as the ratio between the total number of edges incident on all pairs of neighbors of u and the number of edges in a clique formed by the neighbors of u. The *clustering coefficient* of G, denoted by $C(G)$ is the average of clustering coefficients over all nodes. The clustering coefficient of G has values in the range $0 \leq C(G) \leq 1$. There is yet another measure of clustering in graphs, called *transitivity* [NSW02]. *Transitivity* is the ratio

between the number of triangles and the total number of paths of length three.

Average distance: Average distance of G is the mean over all shortest distances between any connected nodes.

Here, we first define Internet graphs and webgraphs. After brief review of results concerning the three salient properties for classical random graphs, "small-worlds," and scale-free random graph processes, we define and study the degree-correlation of the Barabasi-Albert model.

2.2 Internet Graphs

A *physical network* is a collection of interconnected computers, each with its distinct IP address. It can be represented by a connected, undirected graph $G = (V, E)$, with the nodes as computers and the edges as the (physical) communication links (*e.g.*, wire, optical cable). Since G is connected, communication between any arbitrary pair of nodes u and v, takes place through a u, v-path in G. The largest physical network—the Internet—can be modeled on two levels: microscopic and macroscopic. In the *Microscopic Internet graph*, nodes stand for routers and hosts, while edges represent communication links. The *Macroscopic Internet graph* can be thought of as a

contraction of the Microscopic Internet graph: here, each node represents an Internet Autonomous System (which incorporates a number of routers). To simplify the analysis, parallel edges and loops (having negligible influence in modeling propagation) are deleted from the *Macroscopic Internet graph.* Two nodes in the *Macroscopic Internet graph* are adjacent if there is at least one pair of routers (belonging to different autonomous systems) that can communicate. Note that both, the Microscopic and the Macroscopic Internet graphs are undirected.

Faloutsos *et al.* [FFF99] studied particular instances of Internet graphs, and concluded that the degree distribution follows a power-law. In the Microscopic Internet graph, the exponent of the power-law f had value of 2.48, while in the Macroscopic Internet graph, the exponent ranged between $f = 2.15$ and $f = 2.2$ (studies were performed between 1997 and the end of 1998). Govindan and Tangmunarunkit [GT00] mapped the connectivity of nearly 150,000 router interfaces, confirming the power-law exponent of $f = 2.3$. The studies of Yook *et al.* [YJB02] conducted between 1997 and 1999 showed that the Macroscopic Internet graph has clustering coefficient in the range from 0.18 to 0.3 and average distance between 3.70 and 3.77.

2.3 Webgraph

The *webgraph*, representing the WWW, is a directed graph in which nodes represent web pages and directed edges are drawn to indicate the hyperlink relations (the referred URL for a web page is the head and the originating web pages is the tail of the directed edge). As a digraph, the webgraph is characterized by the distribution of in-degrees and the distribution of out-degrees. The sheer order and size of the webgraph implies empirical study only of its subgraphs. Albert *et al.* [AJB99] in the study of the subgraph on roughly 326,000 vertices, found the exponent of the in-degree distribution to be $f = 2.45$ and that of the out-degree distribution to be $f = 2.1$; Kumar *et al.* [KRR99] obtained exponent $f = 2.38$ for the in-degree distribution and $f = 2.1$ for the out-degree distribution of a subgraph on 40 million vertices. The clustering coefficient of the *webgraph* (or its subgraphs) cannot be measured unless the direction of the edges is ignored, since this property is only defined for undirected graphs. Measurement performed by Albert *et al.* [AJB99] demonstrated further that the average distance is 11.2, while subsequent measurements done by Broder *et al.* [BKM00] found that the average distance in a subgraph on 200 million vertices to be 16. Broader *et al.* also presented a fascinating representation of the WWW's macroscopic structure—a bow-tie—composed of three sets: a core, *i.e.*, a strongly-connected component (SCC) composed of mutually-connected nodes, and two sets, IN and OUT, composed of nodes that can only reach (or can only be

reached) from the nodes in the core. Dill *et al.* [DKM02] discovered that the Web graph exhibits fractal properties, *i.e.*, the bow-tie structure appears also on a smaller scale.

2.4 The Salient Characteristics

Reviewing all results from random graph theory is not in the scope of this thesis. Instead, we choose to present results relevant to the three salient characteristic—degree–distribution, clustering coefficient, and average distance—for three classes of random graphs: (1) Erdos–Renyi (classical) random graphs, (2) Watts–Strogatz "small worlds," and (3) Scale–free random graph processes.

2.4.1 Classical Random Graphs

Erdos and Renyi [ER59] employed powerful tools from probability theory to obtain, now classical, results for the space $G_{n,m}$ of all labeled graphs on the set of nodes $V = \{1, 2, \ldots, n\}$ with m edges. The total number of graphs in this space equals $\begin{pmatrix} n\,(n-1)/2 \\ m \end{pmatrix}$. The space $G_{n,m}$ can be turned into *probability space* by assigning equal probability to every realization in $G_{n,m}$. An alternative definition

was given by Gilbert [Gil56]: Given an array of *i.i.d.* Bernoulli random variables $\{X_{ij} : 1 \leq i < j \leq n\}$, with $P(X_{ij} = 1) = p$ and $P(X_{ij} = 0) = 1 - p$, let $G_{n,p}$ be the random graph on $V = \{1, 2, \ldots, n\}$ in which two nodes i and j are adjacent if and only if $X_{ij} = 1$. Informally, an edge in the space $G_{n,p}$ is drawn between any two nodes with probability p. Similarly to $G_{n,m}$, the probability p is a function of n. By taking $m = pn$, one obtains equivalent models.

Theorem 2.4.1. *[Bol85]. Let N_k be the number of vertices of degree k in $G_{n,p}$ with $p = c/n, c > 0$ a constant. Then for $k = 0, 1, \ldots$*

$$P\left((1-\varepsilon)\frac{c^k e^{-c}}{k!} \leq \frac{N_k}{k} \leq (1+\varepsilon)\frac{c^k e^{-c}}{k!}\right) \to 1,$$

as $n \to \infty$.

In other words, Theorem 2.4.1 states that for constant p, the degree distribution of $G_{n,p}$ is approximated by the distribution of the sequence of n *i.i.d.* Bernoulli random variables with probability p and mean pn.

The diameter of the classical random graphs was studied by Bollobas [Bol85] and Chung and Lu [CL01], among others. A general conclusion is that for a given probability p, if $pn/\log n \to \infty$ and $\log n/\log(pn) \to \infty$, then the diameter of $G_{n,p}$ is asymptotic to $\log n/\log(pn)$. The average distance can be obtained by the following

heuristic argument: the number of nodes at distance l from a randomly chosen node grows slower than $\overline{d}^l = (pn)^l$. Taking $n = (pn)^l$, we get $l = \log n / \log (pn)$ [AB02].

Since every edge is chosen independently from the others, the probability that two randomly chosen neighbors of a given node are adjacent is p. Thus, the clustering coefficient $C(G_{n,p}) = p$.

2.4.2 Watts-Strogatz "Small Worlds"

The importance of Watts and Strogatz's model [WS98] lies in the fact that it started a series of mathematical studies of random graphs (other than Erdos-Renyi) defined by simple rules. The first stage of this model is: start with a k-regular graph G of order n, in which: (i) all nodes form a cycle and (ii) two nodes are adjacent if their distance in the cycle on n nodes is no greater than $k/2$. The clustering coefficient of this graph is $\frac{3(k-2)}{4(k-1)}$, which goes to $\frac{3}{4}$ in the limit of large k. If $n = ks$, the diameter of G is s, while the average distance is asymptotically $s/2$ as $n \to \infty$. In the second stage, each edge is deleted independently with probability p, and then all edges are added back at random to obtain the Watts-Strogatz graph.

Watts and Strogatz observed that even for small values of p in the range from 0 to 0.005, the average distance drops down to $O(\log n)$, while the clustering coefficient remains around $\frac{3}{4}$. Barrat and Weigt [BW00] obtained (although not rigorously) an

expression for the degree distribution of Watts-Strogatz model, which resembles the degree distribution for Erdos-Renyi random graphs. The lack of scale-free degree distribution is the principal reason for discarding "small worlds" as a model of real-world networks.

2.4.3 Scale-free Random Graph Processes

The existing models of scale-free random graphs could be divided into two classes: (1) random graph processes and (2) configuration models. The aim of random graph processes is to explain the origin of scale-free degree distribution by running a stochastic graph process with simple rules. On the other hand, configuration models consider the probability space of random graphs on the given scale-free degree sequence.

Molloy and Reed [MR95] were the first to determine the condition for emergence of a giant component in a random graph with a given degree distribution. Their approach was later used by Aiello *et al.* [ACL00] to study the probability space of graphs with a fixed scale-free degree sequence. Generating functions were used in a similar setting by Newman *et al.* [NSW02] to derive results about the average distance and the size of the giant component. In the following sections, we present a brief survey of random graph processes and rigorous results from their analysis.

The most basic random graph process that results in scale-free behavior of the generated graph was given by Barabasi-Albert [BA99]: "... starting with a small number (m_0) of vertices, at every time step we add a new vertex with $m\,(< m_0)$ edges that link the new vertex to m different vertices already present in the system. To incorporate preferential attachment, we assume that the probability Π that a new vertex will be connected to a vertex i depends on the connectivity k_i of that vertex, so that $\Pi\,(k_i) = k_i/\sum k_j$. After t steps the model leads to a random network with $t+m_0$ vertices and mt edges." Thus Barabasi-Albert model allows addition of nodes, one at a time, linked to earlier nodes chosen with probabilities depending on their popularity (a function of the node's degree and additional parameters such as attractiveness or fitness). This idea matches the principle of "the rich gets richer," and has become popular as *preferential attachment rule* [BA99, KR01, DF02].

The informal description of the BA model has two major flaws: (1) The process cannot be started if $m_0 = 1$ and the initial node is isolated, and (2) the choice of the graph induced on the first m_0 nodes determines the outcome of the process: if $m = 1$, and the initial m_0 nodes compose a tree, the outcome of the process is also a tree; however, if the first m_0 nodes compose a forest, the resulting graph would also be disconnected.

To remedy the situation, this random graph process can be inductively defined as follows:

Basis: $G_1 = (V_1, E_1)$, with $V_1 = \{1\}$ and $E_1 = \{(1,1)\}$, *i.e.*, the initial graph is composed of one node, labeled 1, and one loop.

Induction step: Given G_{t-1}, obtain $G_t = (V_t, E_t)$, $V_t = V_{t-1} \cup \{t\}$, $E_t = E_{t-1} \cup \{(t, u_1), \ldots, (t, u_m)\}$, where each u_j, $1 \leq j \leq m$ is chosen from V_{t-1} with probability proportional to its degree, *i.e.*

$$P\left((t,j) \in E_t\right) = \frac{d\left(j\right)}{\sum\limits_{i=1}^{t-1} d\left(i\right)},$$

and $1 \leq i \leq (t-1)$.

Another equivalent definition of this random graph process is given in [BR04, BO04], which allows analysis via its static description—*linearized chord diagram* (LCD) [Sto99]. Consider a fixed sequence of nodes $v_1, v_2, \ldots$. The process $(G_1^t)_{t \geq 0}$ is inductively defined by:

Basis: G_1^1 is composed of one vertex and one loop.

Induction step: Given G_1^{t-1}, G_1^t is obtained by adding a vertex v_t together with a single edge between v_t and v_i, where i is randomly chosen with probability:

$$P\left(i = s\right) = \begin{cases} \dfrac{d_{G_1^{t-1}}\left(v_s\right)}{2t-1}, & 1 \leq s \leq (t-1) \\ \\ \dfrac{1}{2t-1}, & s = t. \end{cases}$$

If the number of added edges, m, from v_t, is greater than one, the process $(\, G_m^t \,)_{t\geq 0}$ is obtained by running $(\, G_1^t \,)_{t\geq 0}$ on the sequence $v_1', v_2', \ldots$; that is, form a graph G_m^t from the graph G_1^{mt} by identifying the nodes $v_1', v_2', \ldots v_m'$ to form v_1, identifying $v_{m+1}', v_{m+2}', \ldots v_{2m}'$ to form v_2, and so forth.

This definition allows the dynamic graph process to be analyzed via its static description—*linearized chord diagram* (*LCD*): The linearized chord diagrams (LCD), with n chords, consist of $2n$ distinct points on the x–axis paired off by semi–circular chords, each chord having one left and one right endpoint. A graph can be obtained from an LCD as follows: starting from the left, identify all endpoints up to and including the first right endpoint reached from node 1. The rest of the nodes are obtained by repeating this process. Finally, the chords from the LCD represent edges in the obtained graph.

Remark 2.4.1. Note that the random graphs, obtained by applying any of the equivalent definitions of the random graph process, are undirected. However, they have natural representation as directed graphs: an undirected edge (i, j) assumes direction from i to j if $i > j$.

Bollobas and Riordan [BR04] analyzed this random graph process and obtained that for fixed $m \geq 1$, $\varepsilon > 0$, and $\alpha = \dfrac{2m\,(m+1)}{(k+m)\,(k+m+1)\,(k+m+2)}$ with probability tending to 1 as $n \to \infty$, $(1-\varepsilon)\,\alpha \leq \dfrac{N_k}{n} \leq (1+\varepsilon)\,\alpha$ for every k in the range $0 \leq k \leq n^{1/15}$, G_m^n is connected and has a diameter $(1-\varepsilon)\log n/\log\log n \leq D\,(G_m^n) \leq (1+\varepsilon)\log n/\log\log n$, and that the clustering coefficient of G_m^n is $\dfrac{m-1}{8}\dfrac{(\log n)^2}{n}$.

Buckley and Osthus [BO04] studied a modification of the preferential attachment by using the approach in [BR04] and obtained that for fixed $m \geq 1$, $a \geq 1$, $\varepsilon > 0$ and $\alpha = (\,a+1\,)\,(\,am+a\,)\,!\begin{pmatrix} k+am-1 \\ am-1 \end{pmatrix}\dfrac{k!}{(\,k+am+a+2\,)\,!}$, $(1-\varepsilon)\,\alpha \leq \dfrac{N_k}{n} \leq (1+\varepsilon)\,\alpha$, k in the range $0 \leq k \leq n^{1/100(a+1)}$. Chung and Lu [CL01] and Cooper and Frieze [CF03] analyzed the preferential rule where nodes and/or edges can be inserted and deleted from the outcome of $(\,G_1^t\,)_{t\geq 0}$. The expected degree sequence in both models is scale-free.

The preferential attachment rule and its variations [BA99, KR01, DF02] are not the only mechanism resulting in scale-free degree distribution. Kumar *et al.* [KRR00] determined another mechanism that not only generates power-law, but also explains the appearance of dense bipartite subgraphs in the Web graph. The basic idea is that a new web page is often made by copying an old one, and then changing some of the links. This model is parameterized by a *copy factor* $p \in (0,1)$ and a constant out-degree $m \geq 1$. At each time step, one vertex is added with m out–going edges.

Generation of out-going edges is a two-stage process: In the first stage, a node—called *prototype*—is chosen uniformly at random from the existing (old) nodes. In the second stage, the destination (node) of the i-th outgoing edge is chosen as follows: with probability p, the destination is chosen uniformly at random from the old node, and with the remaining probability the out-going edge is taken to be the destination of the i-th outgoing edge of the prototype. For a fixed $k > 0$ they obtained that

$$P\left(d\left(u\right) = k\right) = \Theta\left(k^{-\frac{2-p}{1-p}}\right).$$

Dorogovtsev and Mendes [DFS00] and Bollobas [Bol03] analyzed a model in which, at every time step, a node is added and attached to the end vertices of an edge chosen uniformly at random. Since the node sampling based on their degrees could be performed by first choosing an edge uniformly at random and then selecting an end node uniformly at random, this mechanism is equivalent with the preferential attachment rule.

2.5 Degree-correlation of a Scale-free Random Graph

Recent empirical studies of technological and social networks [New89, PVV01] demonstrated correlation among the degrees of adjacent nodes—called *degree-correlation*.

Here, we define and study the degree-correlation for scale-free random graph processes. The methodology we use is similar to that employed in [BR04, BO04]. Our result confirms that node-degrees in the Barabasi-Albert model *are not* correlated (as stated in [New89]).

The Pearson correlation coefficient, r, is a real number, in the range [-1, 1], that expresses the quality of the least squares fitting to a given set of data points (x_i, y_i), $1 \leq i \leq n$. There are two evident problems: (1) how to choose which of the degrees in a pair of adjacent nodes to represent x_i and y_i, and (2) the correlation coefficient should asymptotically hold for any graph generated by the random graph process.

Here, it is more convenient to use the correlation coefficient, r, for two random variables X and Y, written as:

$$r = \frac{cov\,(X, Y)}{\sigma_X \sigma_Y},$$

where $cov\,(X, Y) = E\,[(X, Y)] - E\,[X]\,E\,[Y]$, and (X, Y) represents the joint probability distribution of the random variable X and Y.

Given a random graph G_m^n generated by the random graph process $(\,G_m^t\,)_{t \geq 0}$, consider the two-stage experiment: (1) choose an edge $e = (u, v)$ from G_m^n independently at random, (2) choose one node, say u, incident with e. Let $d\,(u)$ be the

value of X, and $d(v)$ be the value of Y. The probability distribution of the random variable X can be easily derived. Let the number of d-degree nodes be N_d. Since each edge results in two possibilities for successful events, one can obtain the following: $P(X=d) = \dfrac{dP(Z=d)}{\sum_k dP(Z=k)}$, where $P(Z=d)$ is the probability distribution of the random variable representing the degree of a node chosen uniformly at random. Clearly, X and Y have the same probability distribution. The Pearson correlation coefficient can be calculated as:

$$r = \frac{\sum_{d,d'} dd' \left(P(X=d, Y=d') - P(X=d)\, P(X=d')\right)}{\sum_d d^2 P(X=d) - \left[\sum_d dP(X=d)\right]^2}.$$

To make use of this formulation, we need to derive an expression for $P(X=d, Y=d')$.

2.5.1 Joint Probability Distribution

Here, we derive an expression for the joint probability distribution $P(X=d, Y=d')$ (or with shorter notation $P(d,d')$) for degrees of adjacent nodes, writing $\#_m^n(d,d')$ for the number of adjacent pairs of nodes with in-degrees d and d', *i.e.* with total degree of $(m+d)$ and $(m+d')$. We note that for uncorrelated scale-free graphs the

joint probability distribution $P(d, d')$, has the following form:

$$P(d, d') = P(d)\, P(d'),$$

while for correlated scale-free graphs $P(d, d')$ has a more complex form, given by Theorem 2.5.1.

Theorem 2.5.1. *Let $m = 1$ and $(G_1^n)_{n\geq 0}$ be the random graph process defined in Section 2.4.3. Let*

$$\alpha_{d,d'} = \frac{4(d'-1)}{d(d+1)(d+d')(d+d'+1)(d+d'+2)} +$$

$$\frac{12(d'-1)}{d(d+d'-1)(d+d')(d+d'+1)(d+d'+2)},$$

and let $\varepsilon > 0$ be fixed. Then with probability tending to 1 as $n \to \infty$ we have

$$(1-\varepsilon)\,\alpha_{d,d'} \leq \frac{\#_1^n(d,\, d')}{n^2} \leq (1+\varepsilon)\,\alpha_{d,d'}$$

for every $0 \leq d \leq d' \leq n^{1/5}$.

Proof. It turns out that we only need to calculate the expectation $\#_m^n(d, d')$; the concentration result is then given by applying the Azuma-Hoeffding inequality. The

strategy of the proof is as follows: It is enough to consider the case when $m = 1$; the result for general m follows, as mentioned. First, we derive explicitly the joint distribution of D_k and $D_{k'}$, where D_k (resp. $D_{k'}$) is the sum of the first k (resp. k') degrees, assuming $k' > k$. Bollobás and Riordan [BR04] already proved that D_k is concentrated about a certain value. We combine these results to obtain approximately the joint probability $(d_{G_1^n}(v_{k+1}) = d + 1,\ d_{G_1^n}(v_{k'+1}) = d' + 1)$. Summing over k and k' gives the desired result.

Consider first the event $\{D_k - 2k = s\}$, where $0 \leq s \leq n - k$. This is the event that the last $n - k$ nodes of G_1^n send exactly s edges to the first k nodes. This corresponds to a LCD in which the kth right endpoint is $2k + s2k + s$. We shall split the this LCD into left partial LCD L, induced on $\{1, \ldots, 2k + s\}$, and a right partial LCD R, induced on $\{2k + s + 1, \ldots, 2n\}$. Similarly, we arrive at the partial LCDs L' and R', generated by the event $\{D_{k'} - 2k' = s'\}$, where $0 \leq s' \leq n - k'$. Suppose that the left partial LCDs L and L' share j common left unpaired endpoints, where $0 \leq j \leq \min(s, s')$. Consider the event $\{D_k - k = s, D_{k'} - k' = s'|j\}$, the corresponding left partial LCD has exactly

$$\Psi \frac{(2n - 2k' - s')!}{(2n - 2k' - 2s')!} \frac{(2k' - 2k - s + j)!}{(2k' - 2k - 2s + 2j)!} (2n - 2k - 2s - s' + j - 3)!!$$

extensions to a full n-pairing. The term Ψ denotes a rather unilluminating expression that simplifies to $ss' - j$.

This extension of the left partial LCD corresponds to a graph with $d_{k+1} = d + 1$ and $d_{k'+1} = d' + 1$ if and only if $2k + s + d + 1$ and $2k' + s' + d' + 1$ are right endpoints, and each of the $2k + s + 1, \ldots, 2k + s + d, 2k' + s' + 1, \ldots, 2k' + s' + d'$ is a left endpoint. Note that the element paired with $2k + s + d + 1$ must be either one of the s unpaired elements in L or one of the $2k + s + 1, \ldots, 2k + s + d$, and that $s + d - 1$ pairs start before $2k + s + d + 1$ and end after this point. In order for v_{k+1} and $v_{k'+1}$ to be adjacent, it is easy to conclude that $2k' + s' + d' + 1$ must only be paired with one of the unpaired $2k + s + 1, \ldots, 2k + s + d$, and that $s' + d' - 1$ pairs start before $2k' + s' + d' + 1$and end after this point. Since we also have to consider the number j of overlapping unpaired endpoints in the left partial LCDs L and L', we arrive at three cases: (1) $2k + s + d + 1$ chooses among j overlapping left endpoints, $2k' + s' + d' + 1$ chooses among d unpaired left endpoints immediately preceding $2k + s + d + 1$, (2) $2k + s + d + 1$ chooses among $s - j$ non-overlapping left endpoints, $2k' + s' + d' + 1$ makes the same choice as in the previous case, and (3) Each of $2k + s + d + 1$ and $2k' + s' + d' + 1$ chooses one left endpoint from $2k + s + 1, \ldots, 2k + s + d$. Such left partial LCD has exactly

$$\Upsilon \frac{(2n-2k'-s'-d'-1)!}{(2n-2k'-2s'-2d'-1)!} \frac{(2k'-2k-s-d+j-1)!}{(2k'-2k-2s-d+2j-1)!} \cdot$$

$$\cdot \frac{(2n-2k-2s-s'-d-d'+j-1)!}{(2n-2k-2s-s'-2d-d'+j-1)!} (2n-2k-2s-s'-2d-d'+j-4)!!$$

extensions to a full n-pairing. The term Υ denotes a rather unilluminating expression that simplifies to $d(d+s-1)$. Let $M = \lfloor n^{4/5}/\log n \rfloor$, let $k = k(n)$ (resp. $k' = k'(n)$) be any function satisfying $M \leq k(n) \leq n - M$, and let $d = d(n)$ and $d' = d'(n)$ be any two functions satisfying $0 \leq d'(n) \leq d(n) \leq n^{1/5}$. One may obtain:

$$P\left(d_{k+1} = d, d_{k'+1} = d' | D_k - k = s, D_{k'} - k' = s', j\right) =$$

$$(1+o(1)) \left[\frac{2\left(\sqrt{n}-\sqrt{k}\right)^2}{2\left(n-\sqrt{kn}\right)}\right]^{2d'+1} \left[\frac{2\left(k'+k-2\sqrt{kn}+j\right)}{2k'-2\sqrt{kn}+j}\right]^{d+1} =$$

$$(1+o(1)) \left(1-\sqrt{\frac{k}{n}}\right)^{2d'+1} \left(1-\sqrt{\frac{k}{n}}+1-\sqrt{\frac{k'}{n}}\right)^{d+1}.$$

Thus, we arrive at:

$$E\left[\#_1^n\left(d,\,d'\right)\right] \sim \sum_{k'=M}^{n-M}\sum_{k=M}^{n-M}\left(1-\sqrt{\frac{k}{n}}\right)^{2d'+1}\left(1-\sqrt{\frac{k}{n}}+1-\sqrt{\frac{k'}{n}}\right)^{d+1}$$

$$= n^2\int_0^1\left[\int_0^1\left(1-\sqrt{\kappa}\right)^{2d'+1}\left(1-\sqrt{\kappa}+1-\sqrt{\kappa'}\right)^{d+1}d\kappa\right]d\kappa',$$

where $\kappa = k/n$ and $\kappa' = k'/n$. The inner integral yields:

$$\int_0^1\left(1-\sqrt{\kappa}\right)^{2d'+1}\left(1-\sqrt{\kappa}+1-\sqrt{\kappa'}\right)^{d+1}d\kappa =$$

$$\frac{\left(1-\sqrt{\kappa'}\right)^{d}}{\left(3+5d'+2d'^2\right)}\left(\begin{array}{l}\left(1-\sqrt{\kappa'}\right)\left(3+2d'\right){}_2F_1\left(2+2d',\ -d,\ 3+2d',\ -\dfrac{1}{1-\sqrt{\kappa'}}\right)+\\[2ex] +\left(2+2d'\right){}_2F_1\left(3+2d',\ -d,\ 4+2d',\ -\dfrac{1}{1-\sqrt{\kappa'}}\right)\end{array}\right),$$

Integration over κ' gives:

$$E\left[\#_1^n\left(d,\ d'\right)\right]/n^2 \sim$$

$$\frac{(6+4d')\,\Gamma(-2-d)\,\Gamma(3+2d')\,(1+d)\,{}_2F_1(-2-d,2+2d',3+2d',-1)}{(3+10d')\,\Gamma(-d)}$$

$$-\frac{(12+8d')\,\Gamma(-2-d)\,\Gamma(3+2d')\,(2+d)\,(1+d')\,{}_2F_1(-1-d,3+2d',4+2d',-1)}{(3+10d')\,\Gamma(-d)}$$

$$-\frac{(4+4d')\,\Gamma(-1-d)\,\Gamma(4+2d')\,{}_2F_1(3+2d',-1-d,4+2d',-1)}{(3+2d')\,\Gamma(-d)}.$$

Now, by using the Kummer's formula, the theorem follows. □

2.6 Summary

In this chapter, we first define the notions of Macroscopic Internet graph, Microscopic Internet graph, and webgraph. We then briefly survey the empirical results about the properties of large real-world networks. Results about the three salient properties of the scale-free networks: scale-free degree distribution, clustering coefficient, and average distance are reviewed for three classes of random graphs—Erdos-Renyi,

Watts-Strogatz small worlds, and scale-free random graph processes. Finally, we define the notion of degree-correlation, and give a rigorous mathematical result for the degree-correlation of Barabasi-Albert scale-free graphs. The analysis of degree-correlation uses linearized chord diagrams which provide static description of the dynamic Barabasi-Albert model. Finally, we point out that the degree-correlation is a salient-characteristic of scale-free random graphs, and should, therefore, be used in the analysis of stochastic processes (*e.g.* worm propagation) on such random graphs.

CHAPTER 3

WORM PROPAGATION STRATEGIES

3.1 Introduction

In the period from 1979 to 1981 researchers at Xerox PARC built and experimented with worms that were designed to do useful work on the network [SH82]. However, their successor—the infamous Internet worm (released on November 2, 1988)—marked network worms as malicious programs. Analyses [Cen01a, Cen01b] of the recent network worms showed that these intelligent software agents did not perform significant destructive actions (*e.g.*, deleting files, transferring password files, etc.), but propagated quickly across and overwhelmed the networks [Wea02], [SGJ01].

A worm *propagates* by sending a copy of AMMC to a vulnerable host on the network, detected through *scanning* and *probing*. First, *port scanning* determines (by employing a piece of software called *scan*) whether or not a host is reachable over the network. A *scan* request is sent to a specific port of a host (*e.g.*, ICMP_ECHO_REQUEST can be sent using the *ping* utility). A second request, called *probe*, is then used to

detect the services and operating systems running on the host (based on a specific replay to the request) [MSK01]. The size of an AMMC is relatively small (less than a hundred KB), and the scan and probe pieces are negligibly small (a kilobyte and a dozen bytes, respectively) [SPW02]. The code of Sapphire Worm, for example, consisted of only 376 bytes [MPS03]. Thus, the only immediately observable effect of an attack on a network is an increase in the routing-related requests [COP01], as the worm keeps probing different hosts.

Network worms are known to employ malicious actions such as: falsifying addressing information, attack to multiple computers, and simultaneous attacks on a single computer. Unauthorized access on a computer can be gained by *falsifying addressing information* (through sending a request with a forged authorized IP address) in order to bypass a security device (*e.g.*, a firewall). *Attack to multiple hosts* is employed to inflict damage to a network in a shorter time. *Simultaneous attacks on a single host* (*e.g.*, denial-of-service attack) tie up the resources of a computer or a network meant to be used for legitimate purposes. Therefore, if maximum destruction is the intruder's goal, network worms are a cost-effective way to significantly interrupt the functions of the information infrastructure.

3.2 Worms as Multi-agent Systems

According to Nazaro *et al.* [NAW01], malicious mobile-code of a network worm encompasses five (overlapping) components: reconnaissance, specific-attack, command-interface, communications, intelligence, and unused-attack. The *reconnaissance* component comprises the propagation strategy used to identify vulnerable computers. The *specific-attack* component is composed of methods that a network worm uses to gain entry on a computer. For instance, a *buffer overflow* is a method by which the attacker inserts a set of commands in the system program code, thus, masquerading as the operating system. The *command-interface* component allows for controlling the corrupted computers (*e.g.*, a *backdoor* is a feature of a program that enables controlling (accessing) that program without using direct call [PP03]). Sharing information about vulnerable computers is established through the *communication* component, while storing information about corrupted computers is maintained by the *intelligence* component. Methods in the *unused-attack* component employ other vulnerabilities than those in the specific-attack component.

The ability of each copy of the worm to initiate and perform an action without human intervention, naturally lends the autonomous MMCs to be studied as intelligent agents. An *agent* is a conceptual entity (*e.g.*, program, robot) involved in analyzing, structuring, and implementing the processing of a complex problem [Att00]. Franklin

and Graesser's agent-taxonomy includes viruses as a type of software agents [Bra97]. A *software agent* is a program that is capable of initiating an action, has goals and plans to achieve them, communicates with other agents, and responds to events in the environment [Bra97]. The ability to communicate with other agents allows for development of multi-agent systems, in which each agent works on a piece of a complex problem [Woo99].

Moreover, a network worm is an example of a multi-agent system: The propagation of a worm and its survival as a system on the network depend on how each copy of the worm chooses the next node to attack. The copies of the worm are distributed, and have to coordinate with each other, often through messages or pre-defined actions (*e.g.*, all worm copies may be pre-programmed to attack one node at a specific time), in order to inflict more damage to the network.

3.3 Classification of Propagation Strategies

In order to propagate, the worm may use one or a combination of the following propagation strategies: random, local-subnet, topological, hitlist, and permutation [Wea02]. To model defense strategies, we assume that there are security agents deployed on the network. The network on which the worm propagates is modeled by a graph G. At any time step $t > 0$, each node in G is either *occupied* or *unoccupied* by a

copy of the worm or a security agent. The node occupied by the worm will be called *compromised*, while the node occupied by a security agent will be called *guarded*. As the goal of the worm is to maximize the number of compromised nodes, we assume that there can be at most one copy of the worm per compromised node. On a given graph G, representing the network, typically a worm originates at some node y, and propagates according to the steps shown in Figure 3.1.

Algorithm *Worm propagation on a network*

1: A copy of AMMC (worm) is placed on node y
2: The worm selects a *target node* x
3: The worm sends out *scan* and *probe* to node x and looks for the response
4: **if** the response shows that a security agent resides on x **then**
5: **go to** Step 2 (*i.e.*, select another node to compromise)
6: **else if** the response shows that at x there is neither a security agent nor a worm **then**
7: **go to** Step 11 (*i.e.*, compromise node x)
8: **else if** the response shows that a robber already resides on x **then**
9: **go to** Step 2, (*i.e.*, select another node to compromise)
10: **end if**
11: The worm sends a copy of itself and sends the replica to node x (*i.e.*, *compromises node* x)

Figure 3.1: Worm propagation on a graph G

Let at time t, a (copy of) worm reside on node u, and at time $(t+1)$, the worm chooses uniformly at random a node v from a *set of target nodes* (*targets*). In *random selection*, the set of target nodes includes all nodes, except node u. In *local-subnet selection*, only nodes adjacent to u comprise the set of target nodes. For the *topological*

selection, the targets are in the set of nodes with specified topological properties (in relation to node u), denoted by $F(u)$. In the *hitlist selection* strategy, the worm on node u is given a set of targets, called *hitlist*, all of which must ultimately be compromised. In this strategy, after node v has been compromised by node u, the set of targets is evenly divided between the two copies of the worm—one residing on node u, the other on node v. Eventually when the worm's share of targets becomes empty through repeated partitioning of the original hitlist, the local copy of the worm switches to random selection. Finally, in the *permutation selection* strategy, the copy of the worm on node u is given only its first target—node v. Then the next target from node v is node $(v + 1)$, and so on. In all cases, when the number of unsuccessful attacks is goes over some threshold ε, the propagation terminates.

A formal description for the propagation with the random-target selection strategy for the worm residing on node u is given in Figure 3.2. Line 1 initializes Y_{t+1}, the set of compromised nodes by the local copy of the worm up to time $(t + 1)$, with Y_t, the set of compromised nodes by the local copy of the worm to time t. The target (node v) is selected on Line 2. A shortest $u - v$ propagation path, P, in the graph is determined in Line 3. Lines 4 through 10 determine if the node v can be compromised. Line 5 calls a procedure that sends a replica of the worm to node v along the propagation path P. Line 11 returns the set of nodes compromised by the worm on node u up to time $(t + 1)$.

Algorithm *Random propagation*

Input:

u, node on which a copy of the worm resides
w, node from where this copy was sent to node u
$t+1$, current time
Y_t, set of nodes compromised by this worm up to time t
success, indicates if at time t a node has been compromised
counter, number of targets that have not been compromised

Output:

Y_{t+1}, set of nodes compromised by this worm robber up to time $(t+1)$

1: $Y_{t+1} \leftarrow Y_t$
2: **choose** a node v from $V(G) - \{u\}$ uniformly at random
3: $P \quad \leftarrow$ a shortest u, v-path
4: **if** node v is not compromised **then**
5: send_worm(v,u,$t+2$,$\emptyset$, true) //along path P
6: $Y_{t+1} \leftarrow Y_{t+1} \cup \{v\}$
7: $success \leftarrow true$
8: **else**
9: $counter \leftarrow counter + 1$
10: **end if**
11: return Y_{t+1}

Figure 3.2: Random propagation strategy

To obtain the formal description of the local-subnet strategy, Line 2 in Figure 3.2 would be substituted with the following:

choose node v only from among nodes adjacent to node u uniformly at random

Likewise, we would get the procedure for topological strategy (for a predefined topologically related set of nodes to u), if Line 2 in Figure 3.2 is substituted with:

choose a node v from set $F(u)$ uniformly at random

Let H denote the set of targets (hitlist) in the hitlist strategy. The set H_temp stores the initial hitlist, as it is needed in the random strategy when set H becomes empty. We obtain the hitlist strategy, Line 2 in Figure 3.2 should be changed to the following block:

if $|Y_t| = 0$ **then**
  $H\_temp \leftarrow H$
end if
if $H \neq \emptyset$ **then**
  choose a node v from $H - \{u\}$ uniformly at random
  $H \leftarrow$ first half of $H - \{u, v\}$
else
  choose a node v from $V(G) - (\{u\} \cup H\_temp)$ uniformly at random
end if

Finally, one obtains the permutation selection strategy if Line 2 in Figure 3.2 is substituted with:

if *success* = true and *counter* $< \tau$ **then**
  choose a node $v \leftarrow v + 1$
else if *success* = false and *counter* $< \tau$ **then**
  choose a node v from $V(G) - \{u\}$ uniformly at random
else
  stop
end if

3.4 Examples of Worms

Every worm has its unique characteristics in terms of the propagation strategy it employs to spread on the network and the payload it carries. This section provides a description of four worms that caused considerable disruption of Internet services—the variants of the Code Red, Nimda, Slammer, and the Blaster worm.

3.4.1 Variants of Code Red

Code Red started its malicious propagation on July 19, 2001 by exploiting a buffer flow vulnerability in Microsoft IIS server. The Code Red worm propagates as follows: The worm attempts to connect to TCP port 80 on a randomly chosen host assuming that a web server will be found. Upon a successful connection to port 80, the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit the buffer overflow in the indexing service, after which a copy of the worm is sent. If the exploit is successful, the worm copy begins executing on the victim host. The IP

addresses scanned by the Code Red are determined based on the *random propagation strategy* described in Figure 3.2, above.

Code Red II, a variant of its predecessor Code Red, also exploits the buffer overflow vulnerability in Microsoft IIS indexing service. It started propagating on August 6, 2001. The Code Red II worm causes system level compromise and leaves a backdoor on certain machines running Windows 2000, while vulnerable Windows NT 4.0 systems could experience a disruption of the IIS service. The IP addresses scanned by the Code Red II are determined in a probabilistic manner using the *local-subnet strategy*: (1) a random IP addresses with the same first byte as the compromised host is scanned with probability $\frac{1}{2}$, (2) a random IP addresses with the same first two bytes as the infected host is scanned with probability $\frac{3}{8}$, and (3) a random IP addresses in the entire IP space is scanned with probability $\frac{1}{8}$. Thus, Code Red II uses local–subnet scanning. The Trojan horse, installed when the worm is executed, runs every time a user logs in the compromised system, and increases the impact of this worm. As a result of scanning activities, bandwidth denial-of-service has been observed near groups of compromised hosts [Cen01a].

3.4.2 Nimda

The Nimda worm was first seen on September 25, 2001. This worm propagates by multiple mechanisms: from client to client via email, from client to client via open network shares, from web server to client via browsing of compromised web sites, from client to web server via active scanning for (and exploitation of) various Microsoft IIS directory traversal vulnerabilities, and from client to web server via scanning for the back doors left behind by the Code Red II. The payload modifies web documents and certain executable files found on the systems it compromised, and attempts re-sending the compromising e-mail message every 10 days. The compromised host (client) attempts to transfer a copy of the Nimda worm via tftp (69/UDP) to any IIS server that it scans and finds to be vulnerable. The selection of potential target IP addresses is according to the *local-subnet strategy* [Cen01b]: an address with the same first two octets is chosen with probability $\frac{1}{2}$, an address with the same first octet is chosen with probability $\frac{1}{4}$, while a random address is chosen from the entire IP space with probability $\frac{1}{4}$.

3.4.3 Slammer

The SQL Slammer worm (also known as Sapphire worm) started propagating on January, 25, 2003 by exploiting a vulnerability in the Resolution Service of Microsoft SQL Server 2000 [MPS03]. The vulnerability allows for the execution of arbitrary code on the SQL Server host due to a stack buffer overflow. The worm creates packets of 376-bytes and sends them on port 1434/UDP of IP addresses chosen according to the *random propagation strategy*. The propagation of this malicious code has caused varied levels of network degradation across the Internet due to the intensive scanning activities.

3.4.4 Blaster

The propagation of the Blaster worm was initiated on August 11, 2003. It exploits a vulnerability in Microsoft's DCOM RPC interface. Upon successful execution, the worm attempts to retrieve a copy of the file msblast.exe from the attacking host, after successful retrieval runs it, and begins scanning for other vulnerable hosts. To propagate, this worm uses a TCP session to port 135, 139, and 445. The IP address is chosen according to the *local-subnet* propagation strategy [Cen03]: an address with the same first two octets is chosen with probability $\frac{2}{5}$ (the third octet is also

determined from the attacking host according to the *permutation propagation strategy*), while a random IP address is chosen with probability $\frac{3}{5}$. The payload of this worm includes the ability to launch a TCP SYN flood denial-of-service attack against windowsupdate.com.

3.5 Summary

In Section 3.2, we describe the structure of a network worm including five (overlapping) components: reconnaissance, specific-attack, command-interface, communications, intelligence, and unused-attack. Due to the worms' ability to initiate and perform an action without human intervention, we point out that worms can be studied as multi-agent systems. Algorithmic description of five worm propagation strategies: random, local-subnet, topological, hitlist, and permutation is provided in Section 3.3. Local propagation strategies—-local-subnet, topological, and permutation—used by worms presented in Section 3.4 are of particular interest as they strongly depend on the graph structure on which the stochastic propagation process takes place. Understanding the propagation mechanism of a network worm is essential for developing accurate model of propagation.

CHAPTER 4

GRAPH-THEORETIC MODEL OF WORM PROPAGATION AND QUARANTINING

4.1 Introduction

The network on which a worm propagates is a collection of interconnected computers, each with a distinct IP address. Such a network can be represented by a simple, connected, undirected graph G, with the nodes as computers and edges as communication links (*e.g.*, wire, optical cable). The worm is created with the idea to compromise the entire network in the shortest time. A copy of the worm can be viewed as a *robber* placed on a node in G. One must always assume that each robber will propagate copies of itself to all adjacent nodes, whenever it can. Due to cost constraints, it is assumed that only a few copies of the security agents are available. Since at any time only a few sensors alarm the agents, it is reasonable to suppose that only a portion of the available agents are deployed. Agents can be thought of as *cops* that, once

deployed on certain nodes, traverse the graph along the edges. While the robbers do not know the cops' position, the cops know the positions and the propagation strategy of the robbers. Thus, quarantining worm's propagation is an application of the cops-and-robbers game defined in Section 4.3. We believe this game is of fundamental interest in network security as it offers a graph-theoretic framework for studying quarantining strategies.

The chapter is organized as follows: In Section 4.2, we first describe the classical cops-and-robbers game played on a graph, then, present a classification scheme for existing cops-and-robbers games, and, finally, devise a generalization of the classical cops-and-robbers game. In Section 4.3 we present a new results pertaining to the model for quarantining network worms—a class of cops-and-robbers games where the cops have information about the position and strategy of the robbers.

4.2 Classification Scheme for Cops-and-Robbers Games

The classical cops-and-robbers game is played on a simple, connected graph $G = (V, E)$, on n nodes, with c cops and a robber moving in discrete time steps $i = 0, 1, \ldots, t$. To start the game first each of the c cops chooses a node of G, and then the robber is placed on a node of G. Thus, at any time step $i \geq 0$, each node is either *occupied* or *unoccupied* by a player. The node occupied by the robber will

be called *compromised*, while the node occupied by cop will be called *guarded*. For an arbitrary pair of nodes u and v in $V(G)$, a u, v-path will be called *guarded* if it contains a guarded node. Unoccupied nodes to which all paths from the compromised node are guarded will be called *defended*. Unoccupied nodes that are not defended will be called *exposed*. Let at any given time i, X_i denote the set of guarded nodes, Y_i the set containing the compromised nodes, Z_i the set of defended nodes, and F_i the set of exposed nodes, where $X_i \cup Y_i \cup Z_i \cup F_i = V(G)$. The cops and the robber then move alternately, with the cops moving first. A move for the cops consists of each cop either remaining at the same node or sliding along an edge to an adjacent node. A move for the robber is defined analogously. The game is played with perfect information, so that the cops and the robber always know each other's position. The cops win if, after a finite number of moves, one of them can *capture* the robber. We say that the robber is *captured* if there is a cop on the compromised node. The robber wins if it can avoid this situation forever. A *capturing sequence* is the sequence $X = \{X_0, X_1, \ldots, X_t\}$ such that $Y_t \subseteq X_t$. A capturing sequence X will be called *monotone* if the corresponding sequence $Z = \{Z_0, Z_1, \ldots, Z_t\}$ satisfies $Z_i \subseteq Z_{i+1}$, $0 \leq i \leq (t-1)$. Note that when the robber has been captured, all nodes, except for the guarded, are defended, *i.e.*, $F_t = \emptyset$.

Variants of the classical cops-and-robbers game have been applied as models in computer science, operations research, game theory, and control theory. For instance,

these games have been used to model sequential program execution [KP86], planning robot's actions in scenarios of search and rescue [GLL99], and interception of missiles [BO98]. Note that the terms cops-and-robbers, pursuit-evasion, and hunter-and-rabbit are used somewhat synonymously. We adopt the term *cops-and-robbers* as it emphasizes the application of this game to network security.

In this section, we present a classification of existing cops-and-robbers games based on two parameters: Λ_c, cops' information about robber's position, and Λ_r, robber's information about the cops' position. Thus, we represent each of the four possible classes of cops-and-robbers games by a tuple $\langle G, \Lambda_c, \Lambda_r \rangle$. The parameters Λ_c and Λ_r take values 0 or 1, indicating, respectively, no information or complete information. We also discuss the existing variants (within each class) based on two constraints: (1) restriction as to when the robber is allowed to make a move and (2) restrictions on the moves of the cops. The robber that moves only before immediate capture will be called *inert*; without this restriction, the robber will be considered *agile*. Cops are restricted to choose a move from a given subset of the available actions: (A) enter the graph on an arbitrary node, (B) slide along a path with at most s_c edges, or (C) leave the graph. A cop (respectively, a robber) will be said to move with speed s_c (respectively, s_r) if it can traverse a path with at most s_c (respectively, s_r) edges per time step.

Three questions are of particular interest when studying cops-and-robbers games: (1) the structure of the graphs on which a given number c of cops suffice for capture, (2) the minimum number of cops to capture the robber on a given graph G, *i.e.*, the *cop-number*, $c(G)$, and (3) the monotonicity of capturing sequence with $c(G)$ cops on a given graph G.

Notation: Let $d(u, v)$ denote the distance between two arbitrary nodes u and v, and $N(u)$ the set of nodes adjacent to u. When G is a Cartesian product of the graphs G_j, $j = 1, \ldots, k$, $V(G)$ is the Cartesian product of $V(G_j)$, $1 \leq j \leq k$; two nodes $u = (u_1, \ldots, u_k)$ and $v = (v_1, \ldots, v_k)$ are adjacent in the Cartesian product G if and only if $u_j \neq v_j$ for precisely one j $(1 \leq j \leq k)$ and, for this j, (u_j, v_j) is an edge in $E(G_j)$.

4.2.1 Games with Complete Information

In the class of $\langle G, 1, 1 \rangle$ games, the players have complete information about each others' position. Two variants belonging to the class $\langle G, 1, 1 \rangle$ have been studied, wherein: (1) an agile robber moves with unit speed and each cop of unit speed is restricted to choosing action (B) and (2) the inert robber moves with unbounded speed, while cops are restricted in their moves to actions (A) and (C).

4.2.1.1 Agile Robber of Unit Speed, Cops of Unit Speed

First, we survey the results related to the game where an agile robber moves with unit speed and each of the c cops is restricted to only sliding along an edge to an adjacent node. Here, cops have additional knowledge of the robber's strategy. Quilliot [Qui85], Nowakowski and Winkler [NW83] independently determined that on the class of dismantlable graphs one cop can capture the robber. Moreover, Hahn and MacGillivray [HM04] presented a characterization of directed graphs on which c cops can capture the robber. Here, we give the relevant definitions:

Definition 4.2.1. Given a graph G, suppose u and v are two nodes such that $N(u) \cup \{u\} \subseteq N(v)$. The operation of deleting u from G is called a *folding* of G, and we say that u *folds unto* v.

Definition 4.2.2. A graph is called *dismantlable* if there is a sequence of folds that reduces G to a single node.

If G is dismantlable, Algorithm 1 builds a tree T which provides the strategy for the cop. Let $X_i = \{x\}$, $Y_i = \{y\}$. Consider the time step $(t-1)$ just before the robber is captured on such a dismantlable graph G. Since at time step t, $N(y) \cup \{y\} \subseteq N(x)$, the cop captures the robber by moving on y. Otherwise, the cop moves to the node x', adjacent to x in T, closest to y. To avoid the cop, the robber always moves to

Algorithm 1 Tree obtained by dismantling
Input:

G, dismantlable graph

Output:
T, tree obtained from dismantling

```
V(T) ← ∅, E(T) ← ∅
while |V(G)| ≠ 1 do
    M ← ∅
    while there are unmarked u, v ∈ V(G)
    such that N(u) ∪ {u} ⊆ N(v) do
        mark u
        V(T) ← V(T) ∪ {u, v}
        E(T) ← E(T) ∪ {(u, v)}
        M ← M ∪ {u, v}
    end while
    G ← G − M
end while
```

an adjacent node y', furthest from x'. The strategies employed by the cop and the robber are formally described by Algorithm 2.

Maamoun and Meyniel [MM87] proved that one cop can capture the robber (which is not allowed to indefinitely occupy a node) when G is a Cartesian product of two trees T_1 and T_2. Given the guarded node $x = (x_1, x_2)$ and the compromised node $y = (y_1, y_2)$ in $V(G)$, let $d(x_1, y_1)$ denote the distance between the cop and the robber in T_1, and $d(x_2, y_2)$ the distance between the cop and the robber in T_2. The cop moves in G to an adjacent node $x' = (x'_1,\ x'_2)$that minimizes the sum $d(x'_1, y_1) + d(x'_2, y_2)$.The robber moves in G to an adjacent node $y' = y' = (y'_1, y'_2)$that maximizes

Algorithm 2 Strategies on a dismantlable graph
Input:
G, dismantlable graph
i, time step
T, output from algorithm 1
$X_i = \{x\}$, guarded node at i
$Y_i = \{y\}$, compromised node at i
Output:

X_{i+1}, guarded node at $i+1$
Y_{i+1}, compromised node at $i+1$

```
i ← i + 1
// cop's strategy //
if y ∈ N(x) then
    X_i ← {y}
else
    find a node x' ∈ N(x) in T, closest to y
    X_i ← {x'}
end if
// robber's strategy //
find a node y' ∈ N(y) such that d(y', x') = max{d(u, x') : u ∈ N(y)}
// ties are broken arbitrarily //
Y_i ← {y'}
```

$d(x_1', y_1') + d(x_2', y_2')$. We provide a formal description of cop's and robber's strategies in Algorithm 3.

The *retraction strategy* can be used when two cops are in play (*i.e.*, when $c = 2$). To clarify the idea, we give the following definition:

Definition 4.2.3. Given a graph G and a sub-graph H of G, the mapping $\varphi : V(G) \to V(H)$ is called a *retraction* from G into H, if:

(i) $(\varphi(u), \varphi(v)) \in E(H)$ or $\varphi(u) = \varphi(v)$ whenever $(u, v) \in E(G)$ and

Algorithm 3 Strategies on Cartesian product of two trees
Input:

T_1, T_2, trees
G, Cartesian product of T_1 and T_2
i, time step
$X_i = \{(x_1, x_2)\}$, guarded node $x = (x_1, x_2)$ at i
$Y_i = \{(y_1, y_2)\}$, compromised node $y = (y_1, y_2)$ at i

Output:

X_{i+1}, guarded node $x' = (x'_1, x'_2)$ at $i + 1$
Y_{i+1}, compromised node $y' = (y'_1, y'_2)$at $i + 1$

$i \leftarrow i + 1$
$d_1 \leftarrow d(x_1, y_1)$
$d_2 \leftarrow d(x_2, y_2)$
// cop's strategy //
if $N(y) \subseteq N(x)$ **then**
    // robber is captured //
else
    find node $x' \in N(x)$ such that $d(x'_1, y_1) + d(x'_2, y_2)$ is minimized
    $X_i \leftarrow \{(x'_1, x'_2)\}$
end if
// robber's strategy //
find node $y' \in N(y)$ such that $d(x'_1, y'_1) + d(x'_2, y'_2)$ is maximized
$d_1 \leftarrow d(x_1, y'_1)$
$d_2 \leftarrow d(x_2, y'_2)$
$Y_i \leftarrow \{(y'_1, y'_2)\}$

(ii) $\varphi(u) = u$ for each $u \in V(H)$.

Given a shortest path P between two arbitrary nodes u and v, one such mapping from $V(G)$ to $V(P)$ takes $\varphi(w)$ to be the unique node $z \in V(P)$, such that $d(u,z) = \min\{d(u,w), d(u,v)\}$ (*i.e.*, every node $w \in V(G)$, $d(u,w) \geq d(u,v)$, is mapped into node v) [2]. The image $\varphi(y)$ of the compromised node y on the shortest u, v-path P can be thought of as occupied by an *imaginary robber*. Since every path is dismantlable, a single cop can capture the imaginary robber moving along P. The cop then makes the same moves as the imaginary robber. The role of the second cop is to force the robber to move on G. It follows from the definition of retraction that the robber will be captured by the first cop once it decides to enter P.

Here, we give a characterization of graphs on which two cops are necessary and sufficient to capture the robber:

Theorem 4.2.1. *Two cops can capture a robber on a graph G that is not dismantlable but contains a shortest u, v-path P such that each component of $G-P$ is dismantlable.*

Proof. Necessary condition: It follows from the fact that G is not dismantlable and the result from [24]; therefore, more than one cop is necessary for capture.

Sufficient condition: The cops are placed on the given shortest u, v-path P. The robber uses the strategy in Algorithm 2 to avoid the moving onto P. The cops capture the robber in two phases. In the first phase, the cops capture the imaginary robber on

P (see Algorithm 2). Without loss of generality, let $G - P$ have one component, G'. In the second phase, the first cop mimics the moves of the imaginary robber, while the second cop moves according to the strategy in Algorithm 1 on G'. Eventually, the robber will have to enter the path P when it is captured by the first cop. $\square$

Given a graph G that satisfies the conditions of Theorem 4.2.1, Algorithm 4 gives the strategy for the two cops and the moves for the robber.

Algorithm 4 Retraction strategy
Input:

G, graph satisfying conditions of Theorem 4.2.1
φ, retraction to a given shortest u, v-path P
i, time step
$X_i = \{x_1, x_2\}$,guarded nodes at i
$Y_i = \{y\}$, compromised node at i

Output:

X_{i+1}, guarded nodes at $i+1$
Y_{i+1}, compromised node at $i+1$

$i \leftarrow i+1$
// cops' strategy //
if $\varphi(y) \neq x_1$ **then**
  find a node $x_1' \in N(x_1) \cap V(P)$, such that $d(x_1', \varphi(y)) < d(x_1, \varphi(y))$
  $x_2' \leftarrow x_1'$
else
  $x_1' \leftarrow \varphi(y)$
  $x_2' \leftarrow$ Algorithm 2 on G' component of $G - P$, $y \in V(G')$
end if
$X_i \leftarrow \{x_1', x_2'\}$
// robber's strategy //
$y' \leftarrow$ Algorithm 2 on P based on retraction
$Y_i \leftarrow \{y'\}$ φ

Let H be a graph and u be a node of H, such that $H - u$ has no isolated nodes. Using the retraction strategy, Andrea [And86] showed that for any graph G that is not a minor of H, $c(G) \leq |E(H-u)|$. An immediate consequence of this result is that for a planar graph for which K_5 and $K_{3,3}$ are excluded as minors, $c(G) \leq 3$. Aigner and Fromme [AF84] obtained the same result without using the theory of graph minors.

The *shadow strategy* is used to obtain bounds and exact results for the cop-number when the game is played on a graph G—a Cartesian product of given graphs G_j, $1 \leq j \leq k$. We review some definitions:

Definition 4.2.4. A *projection* of G onto G_j is the map $\pi_{G_j} : V(G) \to V(G_j)$ defined as $\pi_{G_j}((u_1, \ldots, u_k)) = u_j$. A player is said to *move in* G_j if its projection onto the other $(k-1)$ factors remains unchanged.

Capturing the projection of the robber on $y = (y_1, \ldots, y_k)$ means placing a cop on $(u_1, \ldots, y_j, \ldots, u_k)$. A cop is said to *shadow* the robber on G_j if at any time step its projection onto G_j is the same as the robber's. Using this strategy, Maamoun and Meyniel [MM87] showed that the cop-number for the Cartesian product on k trees is $\lfloor (k+1)/2 \rfloor$: If each cop c_j, $1 \leq j \leq k$, moves in the Cartesian product of T_{2j-1} and T_{2j} according to Algorithm 3, it will capture the projection of the robber after finite number of steps (provided, the robber is not allowed to remain on one node indefinitely). Each cop c_j, then shadows the projection of the robber. Eventually, the

robber will not have an exposed node to move to. Neufeld and Nowakowski [NN93], [NN98], in addition, obtained bounds for the cop-number of Cartesian, categorical, and strong products of trees and cycles. Bounds for the cop-number of graphs with large girth and Gayley graphs can be found in [Fra87].

Goldstein and Reingold [GR95] proved that for any reflexive, finite, undirected graph G and a fixed integer $K \geq 2$, there is a polynomial-time backtracking algorithm that determines whether $c(G) = K$. On the other hand, they also proved that when K is a parameter, it is EXPTIME-complete to decide whether K cops can win from a given initial position on such a graph. The complexity of the game without giving the initial position has not yet been determined.

4.2.1.2 Inert Robber of Unbounded Speed, Cops only Jump

In the $\langle G, 1, 1\rangle$ game defined by Seymour and Thomas [ST93], the inert robber moves with unbounded speed (note that the robber cannot move through a guarded node). Robbers have additional knowledge about the strategy of the cops. The cops may choose only between actions (A) and (C), *i.e.*, they are restricted to leave and enter the graph (intuitively, the cops must only jump).

On a tree T, two cops suffice to capture the robber: initially, the cops are on some node x, and the robber on node y. Cops move in two alternating phases: In the first

phase, the first cop leaves the graph (this will not cause the inert robber to move). The first cop then enters the game at node y. The robber, with unbounded speed, can move to any node (say u) in the component G' of $G - x$ that contains y (note that the second cop prevents the robber to move to a defended node). In the next phase, the second cop jumps to u. These two alternating phases are iterated until the robber is captured on a leaf node.

Let y be the compromised node at time step i. The set of *compromised* nodes at step $(i+1)$ before the cops "see" the robber can be formally described as: $Y_{i+1} = (V(G) - X_{i+1}) \cup \{u \in V(G) - X_{i+1}$: there is a path from y to u, whose nodes (except y) belong to $V(G) - X_{i+1}\}$.

Seymour and Thomas [ST93] showed that for every graph G on which c cops suffice to capture the robber, there is a monotone capturing sequence X. They proved the monotonicity property by showing that if for a given number of cops the robber has an escape strategy, then there is a collection of sets of nodes that offer a resort to the robber (in the sense that there always exists the possibility for the robber to move from any set of nodes to another one independently of the location of the cops): Given the set of guarded nodes X_i at time step i, a component of $G - X_i$ will be called an *X_i-flap*. The X_i-flap that contains the compromised node will be called a *compromised X_i-flap*. We denote the compromised flap by Y_i. At time step $(i+1)$, either $X_{i+1} \subseteq X_i$ or $X_i \subseteq X_{i+1}$. The robber chooses (if possible) an X_{i+1}-flap,

denoted by Y_{i+1}, such that either $Y_{i+1} \subseteq Y_i$ or $Y_i \subseteq Y_{i+1}$. Notice that the robber does not have a choice if $Y_i \subseteq X_{i+1}$. Therefore, if none of the X_{i+1}-flaps intersects Y_i, the cops have captured the robber. Otherwise the robber chooses from among all X_{i+1}-flaps intersecting Y_i, the one that will become compromised at time step $(i+1)$. Tree-decomposition of the graph G offers one way of formalizing this idea:

Definition 4.2.5. A *tree-decomposition* of the graph G consists of a tree T, and a collection $X = \{ X_u : u \in V(T) \}$ of subsets of $V(G)$, where:

(1) $\bigcup_{u \in V(T)} X_u = V(G)$

(2) every edge of G has both ends in some set X_u, and

(3) for every three nodes u, v, w of $V(T)$, with v lying on the u, w-path,

$$X_u \cap X_w \subseteq X_v$$

The *width of a tree-decomposition* is $\max\{|X_u| : u \in V(T)\} - 1$. The *tree-width* of G is the minimum width of any tree-decompositions of G.

The decomposition of G that realizes the tree-width provides the capturing strategy for the cops—the number of cops necessary to corner the robber on a node equals the tree-width of G. One extra cop is then placed on the compromised node to capture the robber; therefore:

Theorem 4.2.2. *[ST93] The cop-number for the cops-and-robber game with complete information, where the inert robber has additional information about cops' strategy, is equal to the tree-width plus one.*

This result provides information about the structure of the graphs having a large cop-number. Every planar graph is a minor of a large enough square grid. Since every $n \times n$ grid has tree-width n, it is expected that the cop-number for a planar graph will be large.

As determining the tree-width of a given graph G is an NP-hard problem, the problem of determining the cop-number for this variant of the game played on G is also NP-hard.

4.2.2 Games with Complete Information about Cops' Positions

In the class of $\langle G, 0, 1\rangle$ games, the robber has complete information about cops' positions. In this section, we discuss results from four existing games of this class, where: (1) the inert robber moves with speed s_r, cops' moves are restricted to (A) and (C), (2) the agile robber moves with unbounded speed, cops move under restrictions (A) and (C), (3) the agile robber moves with unbounded speed, cops move under

restriction (A), (B) with unit speed, and (C), and (4) the agile robber moves with unbounded speed; cops move with unit speed under restriction (B). It is interesting to note that the strategies for capturing an agile robber, presented in this section, can be used to establish upper bounds on the cop-number in the games presented in Subsection 4.2.1.1

4.2.2.1 Inert Robber of Speed s_r, Cops Only Jump

In the game studied by Dendris *et al.* [DKT97], the inert robber moves with speed s_r just before a cop occupies the compromised node. Similarly to the variant of $\langle G, 1, 1\rangle$ games described in [ST93], the cops are constrained to leave the game and enter the graph on an arbitrary node. Notice that here the cops do not have information about the robber's position. The absence of information for the cops can be described in terms of the set of (*possibly*) *compromised* nodes that need to be inspected by the cops. If Y_i is the set of (*possibly*) *compromised* nodes at step i, we have: $Y_{i+1} = (Y_i - X_{i+1}) \cup \{u \in V(G) - X_{i+1}$: there is a path of length at most s_r, from a node v in $Y_i \cap (X_{i+1} - X_i)$ to u, whose nodes (except v) belong to $V(G) - X_{i+1}\}$. Dendris *et al.* [DKT97] proved that the cop-number for this variant when $s_r = 1$ is equal to the width of the graph plus one.

Definition 4.2.6. A *layout* of a graph G is the n-tuple (ordering) $L = (u_1, \ldots, u_n)$.

The *width of a node* u with respect to a layout L is the number of nodes which are adjacent to and precede u in the layout.

The *width of a layout* L is the maximum width of any node in G. The *width of* G is the minimum width of any layout of G.

Since the width of a graph G is equal to the largest minimum degree of any sub-graph of G (the minimum degree is taken with respect to the sub-graph), there is a polynomial-time algorithm to compute it.

The *width strategy* used by the cops is described as follows: Given a graph G consider the layout $L = (u_1, \ldots, u_n)$ that realizes the width of G. Initially, one cop is placed on node u_1. The following inductive steps are then taken: (i) cop is placed on node u_j, (ii) cops that are on nodes preceding but not adjacent to u_j leave the game (notice that this does not cause the inert robber to move), (iii) cops are placed on the nodes preceding and adjacent to u_j (notice, this prevents the robber from enlarging the set of (possibly) compromised nodes). Clearly, the robber is forced to move further in the layout. Note that the width strategy ignores the whereabouts of the robber and uses a number of cops that equals the width of G plus one. Algorithm 5 provides a description of the width strategy.

For the case when the inert robber moves with a given speed s_r, Dendris *et al.* [DKT97] proved that the cop-number of a given graph G is equal to the tree-width plus one. This result was obtained from the characterization of the tree-width in

Algorithm 5 Width strategy
Input:

G, graph
$L = (u_1, \ldots, u_n)$, layout of G that realizes the width
i, time step
j, rightmost guarded node in L
$X_i = \{x_1, \ldots, x_c\}$, guarded nodes at i

Output:

X_{i+1}, guarded nodes at $i+1$

```
i ← i + 1
if there exists a node x ∈ X_i such that
L(x) < j and (x, u_j) ∉ E(G) then
    X_{i+1} ← X_i − {x}
else if there exists a node x ∈ N(u_j) − X_i such that L(x) < j and (x, u_j) ∈ E(G)
then
    X_{i+1} ← X_i ∪ {x}
else
    j ← j + 1
    X_{i+1} ← X_i ∪ {u_j}
end if
```

terms of the length of the longest cordless cycle of G by using s_r-elimination ordering of G:

Definition 4.2.7. An *s_r-elimination ordering* of the nodes in a graph G, denoted by $\Pi = (u_1, \ldots, u_n)$, specifies the order in which nodes are deleted to obtain a sequence of graphs $(G_1, \ldots, G_{n+1})$.

Given an elimination ordering $\Pi = (u_1, \ldots, u_n)$ and an integer s_r, the graphs generated during an s_r-elimination ordering of the nodes in $V(G)$ according to Π are

recursively defined to be: G_1 is the same as G; the set of nodes for G_{i+1} is $V(G_{i+1}) = V(G_i) - \{u_i\}$, and the set of edges is the set of pairs of nodes $u, v \in V(G_{i+1})$ for which there is a u, v-path in G of length at least s_r such that all its nodes (except u and v) are among $u_1, \ldots, u_i$. Note that G_{n+1} is the empty graph.

Definition 4.2.8. The *s_r-dimension of u_i with respect to* Π is defined to be the degree of u_i in G_i. The *s_r-dimension of* Π is defined to be the maximum s_r-dimension of any node with respect to Π. The *s_r-dimension of* G is the maximum s-dimension of any elimination ordering of G.

The cops use the *elimination-dimension strategy* to capture the robber: Given a graph G and an ordering $\Pi = (u_1, \ldots, u_n)$ of the nodes in $V(G)$, first, a cop is deployed on u_n. Inductively, suppose that cops monotonically inspected the graph induced by the nodes in $u_j, \ldots, u_n$. One cop remains on u_j, the rest of the cops, to the right of u_j, leave the game (this will not cause the inert robber to move). Then, a cop is placed on each preceding node from which there is a path to u_j of length at least s_r. Since the robber is inert, it is forced to move forward in the ordering Π. Therefore, the minimum number of cops necessary to corner the robber on a node is equal to the s_r-elimination dimension of G. Finally, one extra cop is needed to capture the robber. Algorithm 6 formalizes the elimination dimension strategy.

For the case where the inert robber moves with unbounded speed on a graph G, the cop-number of G is again equal to the tree-width plus one. Here, the equality of the

Algorithm 6 Elimination dimension strategy
Input:

G, graph
$\Pi = (u_1, \ldots, u_n)$, ordering of G that
realizes the s_r elimination dimension
i, time step
j, leftmost guarded node in Π
$X_i = \{x_1, \ldots, x_c\}$, guarded nodes at i

Output:

X_{i+1}, guarded nodes at $i+1$

```
i ← i + 1
if there exists a node x ∈ X_i such that L(x) > j then
    X_{i+1} ← X_i − {x}
else if there exists a node x such that L(x) < j and d(x, u_j) ≤ s_r then
    X_{i+1} ← X_i ∪ {x}
else
    j ← j + 1
    X_{i+1} ← X_i ∪ {u_j}
end if
```

tree-width of G and its elimination dimension can be employed again: Since, for any graph, the elimination dimension is equal to its 1-elimination dimension, Algorithm 6 can also provide the strategy for the cops, when the robber moves with unbounded speed. Note that the cop-number for the game with complete information coincides with the cop-number for the game with information about cop's positions, when the robber is inert. This relates the restriction of when the robber is allowed to move with the effect of information that cops have about the opponent's positions and strategy.

4.2.2.2 Agile Robber with Unbounded Speed, Cops Only Jump

When the robber is agile and the cops are restricted to only leave or enter the graph on an arbitrary node, one arrives at the second $\langle G, 0, 1\rangle$ variant (known as *node-search* [KP86]). If Y_i is the set of (*possibly*) *compromised* nodes at step i, the set of (possibly) compromised nodes at step $(i + 1)$ for this game can be formally described as: $Y_{i+1} = (Y_i - X_{i+1}) \cup \{u \in V(G) - X_{i+1}$: there is a path from a node v in Y_i to u, whose nodes (except v) belong to $V(G) - X_{i+1}\}$. Intuitively, for the cops, the robber can be on any of the nodes in Y_i. For this variant, Dendris *et al.* [DKT97] proved, by using the results from [16] and [20], that the cop-number is equal to the path-width plus one. We present some relevant definitions:

Definition 4.2.9. A *path-decomposition* of the graph G consists of a path P, and a collection $X = X = \{X_u : u \in V(P)\}$ of subsets of $V(G)$, where:

(1) $\bigcup_{u \in V(P)} X_u = V(G)$

(2) every edge of G has both ends in some set X_u, and

(3) for every three nodes u, v, w of $V(P)$, with v lying on the u, w-subpath, $X_u \cap X_w \subseteq X_v$

The *width of a path-decomposition* is $\max\{|X_u| : u \in V(P)\} - 1$. The *path-width* of G is the minimum width of any tree-decompositions of G.

The number of cops necessary to corner the robber on a node equals the path-width of G. An additional cop is then placed on the compromised node to capture the robber. Since for any graph G, its node-separation number and path-width are equal [15], the cops can use *node-separation strategy* to capture the robber.

Definition 4.2.10. An edge uv is said to *cross a gap* j, $1 \leq j < n$, in a given layout L of G, if node u precedes and node v succeeds j in L. The *node-separation number of a layout* L is the maximum among the number of edges that cross any gap. *The node-separation number of* G is the minimum node-separation number of any layout.

The *node-separation strategy* is described as follows: Given a graph G, consider a layout $L = (u_1, \ldots, u_n)$ that realizes the node-separation number. We proceed inductively: Initially, the number of nodes that the cops occupy (starting from the first node in the n-tuple) is equal to the node-separation number. Suppose that the cops occupy a cutset between the nodes to the left and right of the gap between u_j and u_{j+1}. By the definition of node-separation number, one of the guarded nodes that precede this gap is not adjacent to any node that succeeds the gap. This cop does not guard any path to the defended nodes, so it can jump (leave and enter the game) on node u_{j+2}. The guarded nodes now form a cutset for the nodes that precede and succeed the gap between u_{j+1} and u_{j+2}. Formal description of the node-separation strategy is given in Algorithm 7.

Algorithm 7 Node-separation strategy
Input:

G, graph
$L = (u_1, \ldots, u_n)$, linear ordering of graph G that realizes the node-separation number
j, rightmost guarded node in L
$X_i = \{x_1, \ldots, x_c\}$, guarded nodes at i

Output:

X_{i+1}, guarded nodes at $i+1$

1: $i \leftarrow i+1$
2: find a node $x \in X_i$ such that $L(x) < j$ and for every $u \in N(x)$, $L(x) \leq j$
3: $X_{i+1} \leftarrow X_i - \{x\}$
4: $X_{i+1} \leftarrow X_i - \{v\}$ $(L(v) = j+1)$
5: $j \leftarrow j+1$

4.2.2.3 Agile Robber of Unbounded Speed, Unrestricted Cops

In the third variant, called *mixed-search*, defined by Bienstock and Seymour [BS91], an agile robber moves with unbounded speed. The cops can either slide to an adjacent node, enter the graph on an arbitrary node, or leave the game. Given a graph G, let H be the graph obtained from G by replacing each edge with two parallel edges. A monotone capturing sequence for c cops in H can be obtained from such capturing sequence with equal number of cops in G, under the restrictions of the second variant.

4.2.2.4 Agile Robber of Unbounded Speed, Cops Restricted to Jump Only After Sliding

In the fourth variant the agile robber moves with unbounded speed. Here, the cops move with unit speed under the restriction (A) and (B). Note that the cop can leave the game only if it has traversed at least one edge. This game, originally known as *edge-search* game, was defined by Breisch (see [Par76]). If Y_i is the set of (*possibly*) *compromised* nodes at step i, the set of (possibly) compromised nodes at step $(i+1)$ for this game can be formally described as: $Y_{i+1} = (Y_i - X_{i+1}) \cup \{u \in V(G) - X_{i+1}$: there is a path from a node v in Y_i to u, whose nodes (except v) belong to $V(G) - X_{i+1}\}\}$. Megiddo *et al.* [MHG88] shows that the problem of computing the cop-number for this variant is NP-hard. LaPaugh [LaP93] proves that if c cops can capture the robber, then there is a monotone capturing sequence X. This result implies the NP-completeness of the corresponding decision problem.

Results related to monotonicity can easily be obtained by using the definition of a crusade [BS91]. For clarification, we present the definition:

Definition 4.2.11. A *crusade* in a graph G, is a sequence $(W_0, W_1, \ldots, W_t)$ of subsets of $E(G)$, such that $W_0 = \emptyset$, $W_t = E(G)$, and $|W_i - W_{i-1}| \leq 1$ for $0 \leq i \leq t$.

Let $\delta(W_i)$ denote the set of nodes which are incident with an edge in W_i and also with an edge in $E(G) - W_i$. The crusade is said to use c cops if $\delta(W_i) \leq k$ for

$0 \leq i \leq t$. By using simple arguments (from sub-modularity of δ), one can prove that there is a c-crusade if and only if there is a progressive c-crusade for which $W_i \subseteq W_{i+1}$, $0 \leq i \leq (t-1)$. Now, given a graph G, let H be the 2-expansion of G (obtained by replacing each edge with two edges in series). A monotone capturing sequence for c cops in H can be obtained from such capturing sequence with equal number of cops in G, under the restrictions of the second variant.

A concept equivalent to a crusade, called *expansion* on a graph was used by Thilikos and Stamatiou [ST00] to prove monotonicity results for the inert robber with unbounded speed for the edge-search and mixed-search games.

4.2.3 Games with no Information about Players' Positions and Strategies

In the class of $\langle G, 0, 0 \rangle$ games, each player has no information about the opponent's position. We will discuss results from the game studied by Spirakis *et al.* [STA95] which draw ideas from the interception game defined in [AKL79]. Here, the cops are divided into two groups: waiting and searching. During the waiting phase, each of the c_w waiting cops performs a random walk starting from the same node until it has visited all of G. Clearly, the waiting cops will be distributed in expected time of order $O(n^3 \log c_w)$, and from then on will remain stationary on the nodes where they ended

the c_w independent random walks. If there is a set Y of nodes in which the robber can move freely (*i.e.* a set does not contain a waiting cop) the expected time that the remaining $(c - c_w)$ cops can capture the robber after at least n random walks is $\min\left\{\left(\frac{n}{c - c_w}\right)^3 \log n,\ \left(\frac{n \log c_w}{c_w}\right)^3\right\}$. Adler *et al.* [ARS02] studied the 0-visibility randomized cops-and-robbers games, and proved that one cop suffices to capture the robber on a general graph in time $O(n \log n)$; however, continuous methods are employed to prove their results. Isler *et al.* [IKK04] shows that two cops suffice to capture the robber in the 1-visibility randomized cops-and-robbers in time $O(n^5)$ (here, a cop performs a random walk until it sees the robber at distance one) by using the methods for dismantlable graphs (as in the game with complete information about opponents' strategies). For the 2-visibility randomized cops-and-robbers, there is a class of random bipartite graphs for which the number of cops required for capture is unbounded.

4.2.4 Games with Complete Information about Robbers' Positions and Strategies

In the class of $\langle G, 1, 0\rangle$ games, the cops have complete information about the robbers, while the robbers have no information about the position of the cops. To our knowledge, there is no variant of the cops-and-robbers game that belongs to this class.

Figure 4.1, below, shows the classification of existing cops-and-robbers games. Figure 4.2 summarizes the results on the three questions of particular interest mentioned in the beginning of Section 4.2.

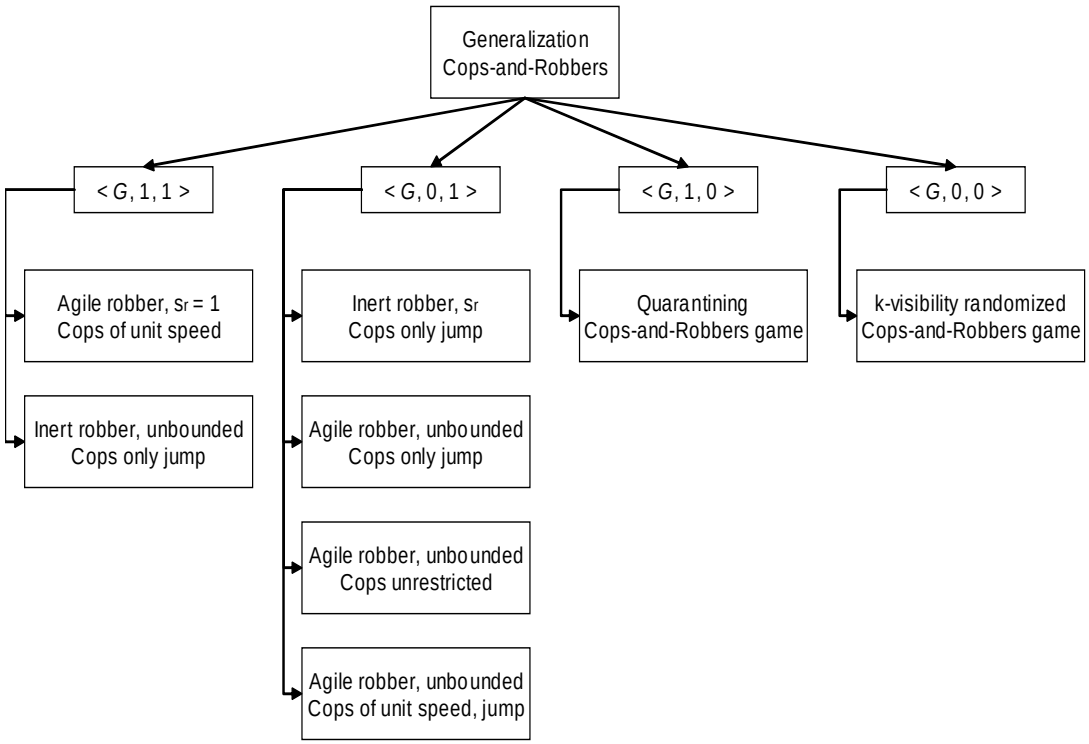


Figure 4.1: Classification of cops-and-robbers games

4.2.5 Generalization of the Classical Cops-and-Robbers Game

The game is played at discrete time steps $i = 0, 1, \ldots, t$. Given a simple, connected graph $G = (V, E)$ on n nodes, a group of c cops of speed s_c try to capture r p-propagating robbers of speed s_r. Note that a player (cop or robber) with speed s can traverse a path with at most s edges per time step. To start the game, at time step $i = 0$, each of the c cops chooses a label in $V(G) \cup \{0\}$, after which the robber is placed on a node of G. A cop can be out of the graph when the game starts, as it is allowed to choose label 0. The cops and the robber then move alternately, with the cops moving first. A move for the cops consists of choosing one of the available actions: (A) enter the graph on an arbitrary node, (B) slide along a path with at most s_c edges, or (C) leave the graph. A move for the p-propagating robber consists of choosing p exposed nodes at distance no greater than s_r, placing a copy of itself on each of the chosen nodes, and then sliding along an unguarded path (of length no greater than s_r) to an exposed node. Thus, depending on the value of p, the number of robbers can increase with time. We assume that at any time there is only one robber per compromised node. The definitions of compromised, guarded, exposed, and defended nodes are analogous to those for the classical cops-and-robbers game.

We say that the robbers are *captured* if there is a cop on every compromised node. Cops win if, after a finite number of moves, they can *capture* the robbers.

The robbers win if they can avoid this situation forever. A *capturing sequence* is the sequence $X = \{X_0, X_1, \ldots, X_t\}$ such that $Y_t \subseteq X_t$. A capturing sequence X is called *monotone* if the corresponding sequence $Z = \{Z_0, Z_1, \ldots, Z_t\}$ satisfies $Z_i \subseteq Z_{i+1}$, $0 \leq i \leq (t-1)$. Note that when $p = 0$, $s_r = 1$, and $s_c = 1$, with cops restricted to action (B), we arrive at the classical cops-and-robbers game. The classification of the cops-and-robbers games presented in Figure 4.1 assumes $p = 0$.

Game	**Class of graphs**	**Cop-number (bounds)**	**Monotonicity**	**Complexity**
Agile robber, $s_r = 1$ Cops slide on edges	Dismantlable	1	Yes	P
	Cartesian product of *k* trees	$\lfloor (k+1)/2 \rfloor$	Yes	P
	Planar	≤ 3	Open	P
	General		Open	EXPTIME-complete: *K* parameter, Initial positions
Inert robber, unbounded Cops only jump	General	$tw(G) + 1$	Open	NP-hard
Inert robber, s_r Cops only jump	General	$tw(G) + 1$	Open	NP-hard
Agile robber, unbounded Cops only jump	General	$pw(G) + 1$	Open	NP-hard
Agile robber, unbounded Cops unrestricted	General	$pw(G') + 1$	Yes	NP-hard
Agile robber, unbounded Cops jump after sliding	General		Yes	NP-hard
Quarantining Cops-and-robbers game	General		Yes	NP-hard [in this paper]

Figure 4.2: Summary of bounds on the cop-number and complexity results for seven variants of cops-and-robbers games

4.3 Cops-and-Robbers Game for Quarantining Network Worms

Here, we define a variant of the generalized cops-and-robbers game that can serve as a model for quarantining the propagation of a network worm. We prove that determining cops' deployment that defends maximum number of nodes is NP-hard.

Our variant of the generalized cops-and-robbers game can formally be defined as follows: Given a graph G and an initial node y (where the first robber is placed), c cops try to quarantine a Δ-propagating robber of unit speed (Δ denotes the maximum degree of G). The cops are restricted to actions (A) entering the graph and (C) leaving the graph. The Δ-propagating robber of unit speed sends copies of itself to all adjacent nodes without information about cops' positions (note that the robber cannot move from its present position, since a new robber is placed on every adjacent exposed node). The cops do not occupy any nodes of the graph up to time step q when they become aware of the robbers' position. To start the game, all cops choose label 0, as to remain out of the graph, after which the robber chooses node y. The robbers move up to time step q, after which the players move alternately, starting with the cops. In a realistic scenario, at any given time step, only p_c of the available cops can be placed on exposed nodes of G. For simplicity, we assume that $c = kp_c$, where k is a positive integer.

The objective of the cops is to quarantine the robbers. The robbers are *quarantined* if there is no robber that can propagate copies of itself to exposed nodes due to presence of cops. The optimization parameter is the number of defended nodes. After the cops have been deployed, they can use the strategies for the game described in Section 4.2.2.4 to capture the robbers.

The formal definition of the optimization problem QUARANTINING OF ROBBERS is given below:

QUARANTINING OF ROBBERS

INSTANCE: Graph G, node $y \in V(G)$ (where the first robber is placed), number of available cops $c \in Z^+$, number of cops $p_c \in Z^+$ deployed per move, and time $q \in Z^+$ when the cops become aware of the robber(s).

PROBLEM: Find a collection of subsets $\{V_1, V_2, \ldots, V_k\}$, $V_j \subseteq V(G)$, $|V_j| \leq p_c$, $V_j \cap V_{j'} = \emptyset$, $1 \leq j \neq j' \leq k$, where for every node $v \in V_j$, $d(y, v) = q + j - 1$, such that $|Z_k|$ is maximized.

The corresponding decision problem is formulated as follows:

QUARANTINING OF ROBBERS

INSTANCE: Graph G, node $y \in V(G)$ (where the first robber is placed), number of available cops $c \in Z^+$, number of cops $p_c \in Z^+$ deployed per move, and time $q \in Z^+$ when the cops become aware of the robber(s), integer K.

QUESTION: Find a collection of subsets $\{V_1, V_2, \ldots, V_k\}$, $V_j \subseteq V(G)$, $|V_j| \leq p_c$, $V_j \cap V_{j'} = \emptyset$, $1 \leq j \neq j' \leq k$, where for every node $v \in V_j$, $d(y, v) = q + j - 1$, such that $|Z_k| \leq K$?

On a rooted tree (T, y) there is a canonical partial order $\leq_{T,y}$ defined as follows: for any two nodes u and v, $u \leq_{T,y} v$ holds if node u lies on the unique path from the root y to node v in T. It is easy to see that the relation $\leq_{T,y}$ is a partial order, *i.e.*, it is reflexive, transitive, and anti-symmetric. A minimal element (respectively, maximal element) is a $u \in T$ such that there is no $v \in T$ with $v \leq_{T,y} u$ (respectively, $u \leq_{T,y} v$). The root is the least element of (T, y) and the leaves are the maximal elements of (T, y), with respect to the partial order $\leq_{T,y}$. If $u \leq_{T,y} v$ holds, then u is called a predecessor of v, while v is a successor of u.

Suppose we are given a rooted tree (T, y),for each $u \in T$ a size $s(u) \in Z^+$, a value $\vartheta(u) \in Z^+$, and an integer knapsack capacity $B \geq \max\{s(u) : u \in V(T)\}$. Let us say that a subset $V' \subseteq V(T)$ is *closed under predecessor* if $v \in V'$ and (u, v) is a directed edge in T imply that $u \in V'$. In the *tree-partially ordered knapsack* problem we wish to find a subset $V' \subseteq V(T)$ which is closed under predecessor,

such that $\sum_{u \in V'} s(u) \leq B$ and $\sum_{u \in V'} \vartheta(u)$ is maximized. This problem is known to be NP-complete, even for the case when $s(u) = \vartheta(u)$ [GJ99].

We prove that the decision problem QUARANTINING OF ROBBERS on trees is NP-complete by restricting it to a modification of the *tree-partially ordered knapsack* problem on *in-trees*: Given an in-tree T, where all edges are directed towards the root, let $l : V(T) \to \{0, \ldots, |V(T)| - 1\}$ be a function assigning to each node a label greater than the labels of its successors, such that $l(y) = 0$. Given an instance of the tree-partially ordered knapsack and a function l, we wish to find a subset $V' \subseteq V(T)$ which is closed under predecessor, such that $\sum_{u \in V'} s(u) \leq B$, $\sum_{u \in V'} \vartheta(u)$ is maximized, and the labels of the maximal elements in V', with respect to $\leq_{T,y}$, are distinct. Observe that, if the function l assigns labels according to the bread-first search of the tree T starting at the root y (and ignoring the direction of the edges), we arrive at the tree-partially ordered knapsack.

Let us now consider the restriction of the QUARANTINING OF ROBBERS on trees:

RESTRICTED QUARANTINING OF ROBBERS

INSTANCE: Tree T, node $y \in V(T)$ (where the first robber is placed), number of available cops $c \in Z^+$, $p = 1$ (one cop deployed per move), cops become aware of the robber(s) at $q = 1$, and integer K.

QUESTION: Is there a collection of sets $\{V_1, V_2, \ldots, V_c\}$, $V_j \subseteq V(T)$, $|V_j| \leq 1$, $V_j \cap V_{j'} = \emptyset$, $1 \leq j \neq j' \leq c$, where for every node $v \in V_j$, $d(y, v) = j$, such that $|Z_k| \leq K$?

Lemma 4.3.1. *RESTRICTED QUARANTINING OF ROBBERS is NP-complete.*

Proof. Given an instance of the modified tree-partially ordered knapsack on an in-tree T where for each $u \in T$, $s(u) = \vartheta(u)$, we construct a corresponding instance of the RESTRICTED QUARANTINING OF ROBBERS as follows: Construct a tree T' on which the game is played from the tree T of the partial order $\leq_{T,y}$ by subdividing every directed edge (u, v) exactly $l(v) - l(u) - 1$ times. This construction can be carried in polynomial time. Exactly $\vartheta(u) - 1$ nodes—predecessors—are added for every node $u \in T$, and each of them is then connected to u by a directed edge. We will denote the nodes of T as *green*, and the added nodes as *red*. Let the number of available cops be B, and the number of defended nodes to be tested is given by K.

First, suppose the desired solution of the modified tree-partially ordered knapsack exists, *i.e.*, there is a subset $V' \subseteq V(T)$ which is closed under predecessor, such that $\sum_{u \in V'} s(u) = B$, $\sum_{u \in V'} \vartheta(u) = K$, and the labels of the maximal elements (with respect to $\leq_{T,y}$) in V' are distinct. From the latter, no more than one cop is placed per level in the tree T', only on green nodes. Consider the sequence of levels on which the cops are placed based on the solution of the modified tree-partially ordered knapsack: the

i^{th} cop in the sequence can be pushed up to its successor on the ith level in T'. This node is one of the red successors before the first green successor. If this were not the case, we would obtain another knapsack solution of bigger value, and thus arrive at a contradiction. Moreover, the number of defended nodes is no less than K.

Now suppose that for a given tree T', rooted at node y, we can find a collection of sets $\{V_1, V_2, \ldots, V_c\}$, $V_j \subseteq V(T)$, $|V_j| \leq 1$, $V_j \cap V_{j'} = \emptyset$, $1 \leq j \neq j' \leq c$, where for every node $v \in V_j$, $d(y, v) = j$, such that $|Z_k| \leq K$. The nodes that are packed in the knapsack are exactly the green ones that are in the sub-trees rooted in nodes of the collection $\{V_1, V_2, \ldots, V_c\}$. Consider the sequence of levels on which the cops are placed: the ith cop in the sequence can then be pushed down to the first green ancestors in T'. The arrangement obtained with this procedure gives a desired solution to the tree-partially ordered knapsack of size no more than B and of value at least K. □

Theorem 4.3.2. *The QUARANTINING OF ROBBERS (decision version) is an NP-complete problem.*

Proof. The theorem follows from the restriction of QUARANTINING OF ROBBERS to trees and Lemma 4.3.1. □

4.4 Summary

As current network-security solutions (*e.g.*, anti-virus software, firewalls, network intrusion-detection systems) offer local protection only from known cyber attacks, devising automated algorithms for quarantining (containing) the propagation of network worms is important for enhancing network security. Cops-and-robbers games offer a framework for devising such algorithms that take into account the characteristics of the graph on which the propagation takes place. Our contribution here is fourfold: (i) a natural generalization of the classical cops-and-robbers game, (ii) a classification of the existing cops-and-robbers games based on two parameters—the information that the two sets of players (cops against robbers) have about each others' position and strategy, (iii) a survey of old results for existing variants of the cops-and-robbers games, and a few new results, and (iv) a new variant of the cops-and-robbers game—a model for quarantining the propagation of cyber attacks. The new variant of the cops-and-robbers game is used to show that the problem of quarantining a network worm is NP–hard.

CHAPTER 5

MODEL OF WORM PROPAGATION ON SCALE-FREE GRAPHS

5.1 Introduction

A worm propagates by employing (local) propagation strategies on various real-world networks, *e.g.*, the Internet, World Wide Web and e-mail networks. Despite ongoing research efforts, there is still no clear understanding of how the underlying scale-free network topology may affect the dynamics of worm spreading when local propagation strategies are employed. Due to the strong analogy between network worms and infectious diseases, epidemiological models have been widely used in modeling propagation. Explicit simulation of worm propagation is, yet, another powerful tool as it allows the effects of the network structure to be considered. Between the extremes of epidemiological approach on a complete graph and individual-based simulation on scale-free networks, it is possible to divide the nodes into groups with similar

characteristics—*e.g.*, degree—and derive a set of equations describing the propagation dynamics. Thus, the network structure can be captured by *stratifying* the population.

Our contribution here is an epidemiological model of worm propagation that makes use of information about the underlying network structure—order, size, degree distribution, and clustering coefficient (transitivity). We compare the results of the model with an individual-based simulation of worm propagation on scale-free Internet graphs obtained from the Oregon Route View project [Pro]. We point out that, although the introduction of the Internet has arguably made the assumption of sparseness no longer valid, the idea of *locality* (especially in the case of analytical modeling of localized propagation strategies) is still applicable, and, therefore, used in our model.

The chapter is organized as follows: In Section 5.2, we present a comprehensive survey and classification of the existing propagation models. The derivation of a pair-approximation Susceptible-Infectious-Susceptible and Susceptible-Infectious-Removed model for worm propagation on scale-free graphs is presented in Section 5.3. Finally, to test the accuracy of our approach, in Section 5.4 we present a comparative empirical study including the numerical solution of the pair-approximation model, the mean-field model, the model of propagation on Erdös-Renyi graphs, and the results of the individual-based simulation.

5.2 Existing Models of Propagation

Due to the strong analogy between network worms and infectious diseases, epidemiological models have been widely used in modeling worm's propagation. Since a worm propagates along the edges of a network, we will use graph-theoretic terms to describe the existing epidemiological models of propagation. Epidemiological models are based on two simplifications [Het00]: (1) At any given time t, each node can be in one of finite number of states, *e.g.*, susceptible, quarantined-susceptible, removed-susceptible, infectious, quarantined-infectious, removed-infectious, and detected. The choice of which states to include in a model depends on the characteristics of the particular worm being analyzed and the purpose of the model; and (2) Translation of the worm transmission mechanism into a probability (rate) of infection. In a similar way, transitions between other states of the model are described by simple probabilities (rates). Epidemiological models can be analyzed analytically or by means of simulation.

The propagation takes place on a graph G with n nodes and m edges. Let $S(t)$ denote the number of susceptible nodes at time t, $Q_s(t)$, be the number of quarantined-susceptible nodes, $R_s(t)$, the number of removed-susceptible nodes, $I(t)$ the number of infectious nodes, $Q(t)$ the number of quarantined-infectious, and $R(t)$ denote the number of removed nodes. The fraction of nodes in a particular state is represented

by the lower case letter. Let β denote the rate at which susceptible nodes are infected. Most models of propagation assume β is constant, averaging out the differences in processor speed, network bandwidth, and location of the infectious node. The existing models also assume that a node cannot be infected multiple times.

Two cases can be modeled based on whether or not control strategies that affect the propagation of the worm are present on the network. The case when control strategies are not in effect is modeled by the Susceptible-Infectious model, while the case when control strategies are present can be modeled by the Susceptible-Infectious-Susceptible or the Susceptible-Infectious-Removed epidemiological model and their variations.

Susceptible-Infectious (SI) model: In this class of models, once a susceptible node becomes infectious, it does not change its state. These models can be used in the study of the *worst-case propagation*, when automated and human counter-measures are not available. Let the average degree of an infectious node be $\overline{d}$, and the fraction of infectious nodes at time t be $i(t)$. The expected number of susceptible neighbors that can be infected by a given infectious node is $\overline{d}(1 - i(t))$. Since there are $I(t)$ infectious nodes, the total rate of newly-infected nodes is $\beta\overline{d}(1 - i(t))\, i(t)$. The general SI model is described by the differential equation (5.2.1):

$$\frac{di\,(t)}{dt} = \beta\overline{d}\,(1 - i\,(t))\,i\,(t)\,, \tag{5.2.1}$$

with boundary conditions: $i\,(0) = \dfrac{I\,(0)}{n} > 0$ and for all $t \geq 0$, $i\,(t) + s\,(t) = 1$. The solution of equation (5.2.1) for the fraction of infectious nodes is the *logistic curve*: $i\,(t) = \dfrac{i\,(0)\ e^{\beta' t}}{1 - i\,(0) + i\,(0)\ e^{\beta' t}}$, where $\beta' = \beta\overline{d}$. The *S-shaped curve* describing the fraction of infectious nodes has three regions: (1) *slow start*, when only few nodes are infected at every time step, (2) *exponential growth*, when the number of newly-infected nodes grows exponentially, and (3) *equilibrium state*, when the number of infectious nodes assumes some value around which it fluctuates steadily.

If the worm propagates on the complete graph on n nodes, K_n, where $\overline{d} = (n-1)$, the model (5.2.1) can asymptotically be written as:

$$\frac{di\,(t)}{dt} = \beta\,(1 - i\,(t))\,I\,(t)\,, \tag{5.2.2}$$

with boundary conditions: $i\,(0) = \dfrac{I\,(0)}{n} > 0$ and for all $t \geq 0$, $i\,(t) + s\,(t) = 1$. One then has that $i\,(t) = \dfrac{i\,(0)\ e^{\beta\,(n-1)t}}{1 - i\,(0) + i\,(0)\ e^{\beta\,(n-1)t}}$. Staniford *et al.* [SPW02] applied model (5.2.2) to fit the data collected by the Chemical Abstracts Services from the propagation of Code Red Worm [Cen01a], and estimated the rate β for Code Red to be

1.8. However, they used the number of scanned nodes, which is much larger than the number of infectious nodes, thus, leading to erroneous conclusions. Weaver [Wea02] and Wagner *et al.* [WDP03] used this model to study four localized propagation strategies: hitlist, topological, permutation, and local-subnet, although we note that the complete graph as underlying topology is inappropriate for studying such localized strategies.

In Erdös-Renyi random graphs with edge-density p, the expected degree of a node is $p\,(n-1)$. The propagation on these graphs can be described as:

$$\frac{di\,(t)}{dt} = \beta p\,(n-1)\,(1-i\,(t))\,i\,(t)\,, \tag{5.2.3}$$

with solution: $i\,(t) = \dfrac{i\,(0)\;e^{\beta\,p(n-1)t}}{1-i\,(0)+i\,(0)\;e^{\beta p\,(n-1)t}}$.

Susceptible-Infectious-Susceptible (SIS) model: In this class of models, an infectious node recovers with some probability, and thus it becomes susceptible again. These models can be used in the study of worm's propagation when some computers are temporarily turned off but are not patched (*e.g.*, the case of Code Red I worm). Let the average degree of an infected node be $\overline{d}$, and the rate with which infectious node recovers be γ. The rate of newly-infected nodes is proportional to the expected fraction of susceptible neighbors, the number of infected nodes, and the probability β.

The rate at which infectious nodes recover is proportional to the number of infectious nodes and γ. The differential equations (5.2.4) describes the general SIS model:

$$\frac{di\,(t)}{dt} = \beta\overline{d}\,(1 - i\,(t))\,i\,(t) - \gamma i\,(t)\,, \tag{5.2.4}$$

with boundary conditions $i\,(0) = \dfrac{I\,(0)}{n}$, and for all $t \geq 0$, $i\,(t) + s\,(t) = 1$. From equation (5.2.4), $\dfrac{di\,(t)}{dt} < 0$ if and only if $s\,(t) < \dfrac{\gamma}{\beta\overline{d}} = \delta$. Thus, the worm "dies out" if the initial fraction of susceptible nodes is below the *epidemic threshold* $\dfrac{\gamma}{\beta\,\overline{d}}$. The solution of (5.2.4) gives a functional form for the fraction of infectious nodes: $i\,(t) = \dfrac{(1-\delta)\,i\,(0)}{i\,(0) + (1 - \delta - i\,(0))\;e^{-(\beta' - \gamma)t}}$, where $\beta' = \beta\overline{d}$. If the worm propagates on the complete graph on n nodes, K_n, where $\overline{d} = (n-1)$, the model (5.2.4) can asymptotically be written as [Sol90]:

$$\frac{di\,(t)}{dt} = \beta\,(1 - i\,(t))\,I\,(t) - \gamma i\,(t)\,, \tag{5.2.5}$$

with solution $i\,(t) = \dfrac{(1-\delta)\,i\,(0)}{i\,(0) + (1 - \delta - i\,(0))\;e^{-(\beta(n-1) - \gamma)t}}$.

Solomon studied a modification of model (5.2.5) where the rate γ is a weighted average of the rate γ_1 (describing computers not running anti-virus software), for

the fraction of infectious nodes, and the rate γ_2 (describing computers running the most recent version of anti-virus software), for the fraction of susceptible nodes, *i.e.*, $\gamma = \gamma_1 i(t) + \gamma_2 (1 - i(t))$. With this modification Solomon found that the necessary effectiveness of the anti-virus software (described by the probability γ) should be 0.5 in order to stop the propagation before it achieves exponential growth.

Kephart *et al.* [KW91, Kep94] employed model (5.2.4) to study the effects of three topologies on the propagation of *viruses*: Erdös-Renyi random graphs, regular lattices of degree eight, and hierarchically-clustered random graphs. For the Erdös-Renyi random graphs with $\overline{d} \geq 5$, simulation results coincide with the predictions of the model. The simulation study of propagation on 100–by–100 lattice demonstrates quadratic growth, in contrast with the exponential growth characteristics for the complete graph and Erdös–Renyi graphs. The hierarchically-clustered random graphs are generated as follows: given a rooted tree of height h, in which every node has a degree $(d+1)$ (*i.e.*, it has d successors), the nodes of the graph are the leaves of the tree. Two nodes, u and v, are made adjacent with probability $P(h(w))$ proportional to the height of node w—the first common ancestor of nodes u and v. In his simulation, Kephart used $P(h(w)) = \alpha p^{h(w)}$, where parameter p is used to control the degree of localization (when p tends to 0, the graph is composed of isolated nodes, while when p approaches 1, the topology of the hierarchically-clustered random graph is asymptotically that of the Erdös-Renyi random graphs). Here, the propagation exhibits

sub-exponential growth. Further simulation studies conducted by Kephart [Kep94] shows that sparsely-connected (random) graphs inhibit the propagation.

Previous models are limited in their accuracy due to their simplistic treatment of timing factors, such as *infection delay*—the length of time between the instant of worm's arrival at a node and the instant when this node becomes infectious to its neighbors. Model (5.2.4) could be altered to incorporate the infectious delay, ε, as follows [WW03]:

$$\frac{di\left(t\right)}{dt} = \beta\overline{d}e^{-\gamma\varepsilon}\left(1 - i\left(t\right)\right)i\left(t - \varepsilon\right) - \gamma i\left(t\right), \tag{5.2.6}$$

where $i\left(t - \varepsilon\right) = 0$ for $t < \varepsilon$. At time $t \geq \varepsilon$, the fraction of infectious nodes is the same as the fraction of infectious nodes at time $(t - \varepsilon)$, since all nodes infected between $(t - \varepsilon)$ and t are delayed. The term $e^{-\gamma\varepsilon}$ accounts for the transfer of a node from infectious to susceptible state during the delay period. Equation (5.2.6) belongs to the class of non-linear delayed differential equations, which can be solved under the assumption $i\left(t - \varepsilon\right) = i\left(t\right)$. Wang *et al.* [WW03] support their analytical solution with simulation similar to that of Kephart and White [KW91], and show that the epidemic threshold depends not only on the average degree, but also on the infection delay. In addition, Kim *et al.* [KRD04] performed a simulation study of

the propagation on a subgraph of the Internet, using a constant delay equal to the average round-trip time obtained from real-life traffic.

Pastor-Satorras *et al.* [PV02] modified model (5.2.4) to study the effects of the scale-free Barabasi-Albert topology on the propagation with probability of recovery $\gamma = 1$. Since a scale-free degree distribution is not concentrated around its mean value, the model must include differential equation for every group of nodes of degree k:

$$\frac{di_k(t)}{dt} = \beta k \left(1 - i_k(t)\right) \Theta \left(\{ i_k(t) \}_{k=\delta}^{\Delta} \right) - i_k(t), \tag{5.2.7}$$

where $\Theta \left(\{ i_k(t) \}_{k=\delta}^{\Delta} \right)$ describes the probability that a susceptible node of degree k is adjacent to an infectious node. For a scale-free network, the probability that an edge is incident on a node of degree k is $kP(k)/\overline{d}$. The average probability that an edge is incident on an infectious node is then $\Theta(t) = \frac{1}{\overline{d}} \sum_{k=\delta}^{\Delta} kP(k)\, i_k(t)$. The conclusion of this model is that scale-free topologies do not have epidemic threshold. The authors also argued that the cut-off in the scale-free distribution forces a non-zero epidemic threshold. We point out that the result of this study is limited to scale-free topologies without degree-correlations. Contrary to this result, the simulation study

of Eguiluz and Klemm [EK02b] demonstrates that in the so-called *structured scale-free networks*, where adjacent nodes share large number of common neighbors, there exists a non-zero threshold even in the limit of large n.

While the results of the presented studies are valuable, a model in which node that has recovered and is no longer susceptible could better approximate the realistic propagation of a worm when human counter-measures are in place.

Susceptible-Infectious-Removed (SIR) model and its variations: In this class of models, an infectious node can be removed (*i.e.*, it can no longer spread the worm). This model can be used to study the effects of software patching and traffic blocking on the propagation. At any time t, a node can be susceptible, infectious, or removed. Let γ be the rate with which infectious nodes are removed. Using analogous arguments as in the previous section, the general SIR model can be written as:

$$\begin{aligned} \frac{di(t)}{dt} &= \beta\overline{d}\,(1 - i(t))\, i(t) - \gamma i(t)\,, \\ \frac{dr(t)}{dt} &= \gamma i(t)\,, \end{aligned} \tag{5.2.8}$$

with initial conditions: $i(0) = \dfrac{I(0)}{n} \geq 0$, $r(0) = \dfrac{R(0)}{n} \geq 0$, and for all $t \geq 0$, $i(t) + s(t) + r(t) = 1$. The epidemic threshold for SIR models is analogous to the one in SIS models. Zou *et al.* [ZGT02] used a modification of the system (5.2.8) to determine the

effect of human counter-measures (on removing both susceptible and infectious nodes) and the decreasing rate $\beta(t)$. This so-called *two-factor* model assumes complete graph as underlying topology, and a constant fraction of the removed-infectious nodes at any time t:

$$\begin{aligned}
\frac{di(t)}{dt} &= \beta(t)\left(1 - r(t) - r_s(t) - i(t)\right) i(t) - \frac{dr(t)}{dt}, \\
\frac{dr(t)}{dt} &= \gamma i(t), \\
\frac{dr_s(t)}{dt} &= \mu\left(1 - r(t) - r_s(t) - i(t)\right)\left(r(t) + i(t)\right), \\
\beta(t) &= \beta(0)\left(1 - i(t)\right)^{\eta}.
\end{aligned} \tag{5.2.9}$$

It is, however, unclear how the parameters have been chosen in order to fit the data from the Code Red I worm propagation.

Boguna *et al.* [BPV03] studied the SIR model, with the probability $\gamma = 1$, on scale-free topologies. Using the notation introduced in previous sub-section, the model can be formulated as follows:

$$\frac{di_k(t)}{dt} = \beta k (1 - i_k(t)) \Theta \left(\{ i_k(t) \}_{k=\delta}^{\Delta} \right) - i_k(t),$$

$$\frac{dr_k(t)}{dt} = \gamma i_k(t), \tag{5.2.10}$$

which can be solved if one assumes that $i(0)$ is very small in the beginning of the propagation, to obtain an epidemic threshold with value $\frac{E[d(v)]}{E\left[d(v)^2\right] - E[d(v)]}$. Since for the power-law degree distribution with exponent $2 < f < 3$, $E\left[d(v)^2\right] \to \infty$, an epidemic threshold *does not exist*, in striking contrast with the classical SIR. Pastor-Satorras and Vespignani [PV02] conducted simulation study to investigate the effects of node-immunization (*i.e.* node-removal) on the propagation, before the worm is introduced in the network. They demonstrated that random immunization is inefficient in slowing down the propagation; however, immunization targeted at nodes of highest degrees can significantly inhibit the growth of propagation. While the latter result seems interesting, the authors argue that detecting nodes of high degrees in scale-free networks is a difficult problem. We first note that $\Theta \left(\{ i_k(t) \}_{k=\delta}^{\Delta} \right)$ is a non-increasing function, since an infectious node would have an increasing portion of recovered nodes in its neighborhood, which limits the propagation. Therefore, an accurate model of propagation should also include correlations between states of nodes which arise from the random propagation process.

Similarly, the simulation study of Wang*et al.* [WKE00] examines the effects of immunization of nodes on the propagation on two topologies: rooted trees and clustered networks (composed of cliques inter-connected with small number of edges). The simulation's parameter is the *propagation fan out*—number of nodes to which the worm can send replicas at each time step. The time needed for the worm to propagate from one node to another is assumed to be one time tick. The first set of simulation is conducted on networks where no immunized nodes exist to determine the number of times a node is re-infected (called *re-infection count*). Two types of immunization are simulated—random and selective. Random immunization performs better on rooted trees as there is only one path between any two nodes; thus, it is possible to cut off an entire sub-tree of the network, which is not the case with the clustered network. In the analysis of the selective immunization in rooted trees, nodes with highest re-infection counts are chosen (note, these nodes coincide with nodes with largest degrees). In the case of clustered networks, two strategies are used: first based on the re-infection count, and second on the weighted sum of the inter-cluster and inner-cluster degrees for every node. The first strategy is able to contain the propagation, but results in a higher propagation rate. The second could slow down the propagation rate, but is unable to contain the propagation.

The principal disadvantage of the studies in [WKE00] and [PV02] is that immunization is static, *i.e.*, a fraction of nodes is immunized before the worm starts

propagating. In reality, the counter-measures should be dynamic in nature to play important role in slowing down the propagation of the worm.

Susceptible-Infectious-Detected-Removed (SIDR) model: This model was analyzed by Williamson *et al.* [WL03] in order to determine the effectiveness of the behavior-blocking approach called virus throttling [Wil02]. Virus throttling is an automatic mechanism for slowing a worm's propagation. Here, a node can be in one of the four states: susceptible, infectious, detected (in which the virus has been detected and cannot actively spread further), and removed. The model assumes complete graph as underlying topology. The model involves two stages: in the first stage, prior to the release of the virus signature, nodes progress from susceptible to infectious state at some rate β. In the second stage, after some time from the start of the propagation, the virus is detected with some probability γ. Two quantities are studied: the number of infectious nodes and the duration of propagation. The model incorporates virus throttling by dividing the nodes into two groups—throttled and un-throttled. If a throttled node is infected, it does not spread the virus, and immediately enters the detected state. The result shows that when more than half of the nodes have throttles, even a delayed distribution of the worm signature will result in a small outbreak.

Susceptible-Infectious-Removed-Susceptible (SIRS) model: Wang *et al.* [WW03] used a modification of SIS model (5.2.4) to study the node's vigilance against infection: Once an infectious node is removed, it remains in this state for a length of time ν, called *vigilance period*, after which the removed node becomes susceptible again. Here, the susceptibility of a node is modeled via a parameter ϕ that takes values between 0 (indicating complete susceptibility) and 1 (indicating immunity). The model is described by the non-linear delay differential equation (5.2.11):

$$\frac{di\left(t\right)}{dt} = \beta\overline{d}\left(1 - i\left(t\right) - \int_{t-\nu}^{t} i\left(t\right)\right) i\left(t\right) - \gamma i\left(t\right) \tag{5.2.11}$$

whose solution shows that the number of infectious nodes decreases as the vigilance period increases. It is worth noting the node's vigilance has no impact on the epidemic threshold.

Compartmental epidemiological models: Compartmental epidemiological models can are used with stratified population. The topology in this models is the *macroscopic Internet graph*, where every node represents a dense region—Autonomous System (AS). These models can be used to study intra-AS propagation, with the assumption that within an AS (with n_j nodes) the worm propagates as on a complete graph K_{n_j}. The infectious attempts can then be modeled as being external or internal

to an AS. If the macroscopic Internet graph has k nodes, the SI compartmental model can be written as:

$$\frac{di_j(t)}{dt} = \left[\sum_{l=1}^{k} \beta \frac{n_l}{N} i_l(t)\right](1 - i_j(t)), \tag{5.2.12}$$

where $1 \leq j \leq k$. Here, the parameter N denotes the total number of IP addresses. Serazzi *et al.* [SZ04] used model (5.2.12) to derive equations for the bandwidth consumption at each node. For the SIR compartmental model, Liljenstam *et al.* [LNB03] obtained:

$$\begin{aligned} \frac{di_j(t)}{dt} &= \left[\sum_{l=1}^{k} \beta \frac{n_l}{N} i_l(t)\right](1 - i_j(t)) - \gamma i_j, \\ \frac{dr_j(t)}{dt} &= \gamma i_j(t), \end{aligned} \tag{5.2.13}$$

where $1 \leq j \leq k$. Liljenstam *et al.* [LNB03] used model (5.2.13) to study the destabilizing effects of worm propagation on the network infrastructure, since the compartmental approach allows for inclusion of limited details about communication protocols. In this simulation study, the scan traffic is modeled by using a combination of the average scan rate, individual infection rates, and size of address space for each AS.

Discrete-time approximation models: Chen *et al.* [CGK03] developed a deterministic approximation model of propagation on a complete graph K_n. If σ is the average scanning rate, with the assumption that the total number of nodes is 2^{32}, the average number of newly–infected nodes at step $(t+1)$ is $(S(t) - I(t)) \left[1 - (1 - 1/2^{32})^{\sigma I(t)}\right]$. If the probability of removal is γ, in the next time step $\gamma I(t)$ nodes will become susceptible. Thus, the propagation can be described by a system of recurrences for the number of infectious and susceptible nodes.

5.3 Pair-approximation Model on Scale-free Networks

The existing epidemiological models on scale-free graphs [PV02, ZGT02] do not explicitly give the system of differential equations for the propagation dynamics. The comparative studies include either simulation of worm's propagation on a macroscopic level or a system of differential equation for propagation on Erdös-Renyi and regular graphs. Thus, in all models described in Section 5.2, it is not evident how the scale-free topology might affect the propagation.

Here, we develop a realistic model of worm propagation by using the salient features of the underlying scale-free graphs (models of real-world networks). Cast in the Susceptible-Infectious framework, our model can be used to study the worst-case propagation and determine the optimal time for undertaking preventive action. On

the other hand, cast in the Susceptible-Infectious-Removed framework, this model can be used in the study of near-optimal control strategies against network worms.

Our model of worm's propagation belongs to the class of *pair-approximation network models*. The benefit of this class of models is that it incorporates the spatial structure that the existing epidemiological models of propagation ignore. A survey of pair-approximation models is given by Rand [Ran99]. In the pair-approximation model, the variables are the fractions of pairs of nodes in certain states. Usually, these equations contain higher-order correlations (*e.g.*, triples of nodes in certain states) which are approximated by the lower-order correlations. For most part, previous work on pair-approximation models describes processes on regular-lattices. Our model extends the work by Earnes and Keeling [EK02a] (for triangle-free networks) and Bauch [Bau02] (for dynamic partnerships), and makes pair-approximation applicable to various scale-free topologies.

Next, we present the derivation of the system of differential equations describing the propagation in the Susceptible-Infectious-Susceptible (SIS) framework. Let $N(u)$ is the neighborhood of a node u, $p_t(i_u)$ is the probability that, at time t, node u is infectious, and $p_t(s_u, i_v)$ is the joint probability that two adjacent nodes u and v are susceptible and infectious, respectively. The time evolution of the state of a single node in the SIS epidemic process can be written in the following form:

$$\frac{dp_t\left(i_u\right)}{dt} = \beta \sum_{v \in N(u)} p_t\left(s_u, i_v\right) - \gamma p_t\left(i_u\right), \tag{5.3.1}$$

$$p_t\left(s_u\right) + p_t\left(i_u\right) = 1.$$

One can also develop an equation for the time evolution of $p_t\left(s_u, i_v\right)$ which in turn involves higher-order correlations. The SIS epidemiological model neglects the higher-order correlations with the assumption that $\left(s_u, i_v\right) = p_t\left(s_u\right) p_t\left(i_v\right)$. In our approach, $p_t\left(i_u\right)$ and $p_t\left(s_u, i_v\right)$ are kept as variables of interests while the higher-order correlations are expressed in terms of these variables. The time evolution of $p_t\left(s_u, i_v\right)$ can be derived by using the Kolmogorov forward equation:

$$\begin{aligned}\frac{dp_t\left(s_u, i_v\right)}{dt} = & -\left(\beta + \gamma\right) p_t\left(s_u, i_v\right) - \beta \sum_{w \in N(u)-v} p_t\left(i_w, s_u, i_v\right) \\ & + \beta \sum_{w \in N(v)-u} p_t\left(s_u, s_v, i_w\right) + \gamma p_t\left(i_u, i_v\right).\end{aligned} \tag{5.3.2}$$

Let Λ_a be the set of integers representing degrees of nodes adjacent to nodes of degree a, and $[ab]$ be the number of edges incident on nodes of degrees a and b. To avoid lengthy derivations for every pair of nodes in different states, we will use X, Y, and Z to denote a node-state (*i.e.*, susceptible and infectious). Given a node u of degree a and a node v of degree b, define

$$p_t\left(X_a, Y\right) = \frac{1}{a} \sum_{d(u)=a, d(v)\in\Lambda_a} p_t\left(x_u, y_v\right)$$

and

$$p_t\left(X_a, Y_b, Z\right) = \frac{1}{[ab]} \sum_{d(u)=a, d(v)=b, d(w)\in\Lambda_b} p_t\left(x_u, y_v, z_w\right).$$

Notice that $p_t\left(X_a, Y\right) = \sum_{k\in\Lambda_a} P_t\left(X_a, Y_k\right)$ and $p_t\left(X_a, Y_b, Z\right) = \sum_{k\in\Lambda_b} p_t\left(X_a, Y_b, Z_k\right)$. Furthermore, let $E\left[\, X_a \,\right]$ denote the expected number of nodes of degree a in state X, $E\left[X_a Y_b\right]$ the expected number of pairs of nodes of degree a, in state X, adjacent to nodes of degree b, in state Y, and $E\left[X_a Y_b Z_c\right]$ denote the expected number of triples where a node of degree b, in state Y, is adjacent to a node of degree a and a node of degree c, in state X and Z, respectively. By multiplying equation (5.3.1) with n_a, the number of nodes of degree a in the graph G, one can obtain the following equation:

$$\frac{dE\left[I_a\right]}{dt} = \beta \sum_{k\,\in\,\Lambda_a} E\left[S_a I_k\right] - \gamma\, E\left[I_a\right], \tag{5.3.3}$$

where $E\left[S_a\right] + E\left[S_b\right] = n_a$. Similarly, one can transform equation (5.3.2) to obtain:

$$\frac{dE\left[S_a I_b\right]}{dt} = -\left(\beta+\gamma\right)\, E\left[S_a I_b\right] - \beta \sum_{k\,\in\,\Lambda_a} E\left[I_k S_a I_b\right] + \beta \sum_{k\,\in\,\Lambda_b} E\left[S_a S_b I_k\right] + \gamma E\left[I_a I_b\right]. \tag{5.3.4}$$

In epidemiological models any change in the state of a node is dependent on the states of its neighbors [Kee99, Ran99], as susceptible node with many infectious neighbors is likely to become infectious. Given a node u of degree a in state X, let $Q_u(Y|X)$ denote the number of its neighbors of degree b that are in state Y. The expected value of $Q_u(Y|X)$ over all nodes of degree a in state X is then $\overline{Q_u(Y|X)}$, calculated as

$$\overline{Q_u(Y|X)} = \frac{[X_a Y_b]}{[X_a]} + error\left(Q_u(Y|X)\right).$$

Assuming that the error is multinomial (supported by the individual-based simulation in Section 5.4), one can obtain the following result for the triples $[X_a Y_b Z_c]$ (for simplicity, the symbol of expectation is neglected):

$$[X_a Y_b Z_c] = \sum_{u:d(u)=b} Q_u(X_a|Y_b) Q_u(Z_c|Y_b) =$$

$$= \sum_{u:d(u)=b} \left(\frac{[X_a Y_b]}{[Y_b]} + error\left(Q_u(X_a|Y_b)\right) \right) \left(\frac{[Y_b Z_c]}{[Y_b]} + error\left(Q_u(Z_c|Y_b)\right) \right) =$$

$$= \frac{[X_a Y_b][Y_b Z_c]}{[Y_b]} + \sum_{u:d(u)=b} error\left(Q(X_a|Y_b)\right) error\left(Q(Z_c|Y_b)\right) =$$

$$= \frac{b-1}{b} \frac{[X_a Y_b][Y_b Z_c]}{[Y_b]}.$$

Let φ_{abc} denote the transitivity among nodes of degree a, b, and c, *i.e.*, the ratio of the number of triangles to the number of connected triples whose nodes are of degree a, b, and c. To approximate the third moment $[X_a Y_b Z_c]$, one has to use the definition of multiplicative moments of two variables (derived from much simpler form for regular lattice in [Kee99, Ran99]):

$$cor\left(X_a, Z_c\right) = \frac{n_a n_c}{[ac]} \frac{[X_a Z_c]}{[X_a]\,[Z_c]}$$

Finally, since the transitivity φ_{abc} gives the probability that three nodes of degrees a, b, and c form a triangle in G, the approximation for the number of triples $[X_a Y_b Z_c]$ can be written as:

$$[X_a Y_b Z_c] = \frac{b-1}{b} \frac{[X_a Y_b]\,[Y_b Z_c]}{[Y_b]} \left((1-\varphi_{abc}) + \varphi_{abc} \frac{n_a n_c}{[ac]} \frac{[X_a Z_c]}{[X_a]\,[Z_c]} \right). \tag{5.3.5}$$

Similarly, one can derive formulae for the other second moments and appropriate approximation of the third moments to obtain the following pair-approximation for the SIS framework:

$$\frac{d\,[I_a]}{dt} = \beta \sum_{k\in\Lambda_a} [S_a I_k] - \gamma\,[I_a]\,,$$

$$\frac{d\,[S_a]}{dt} = \gamma\,[I_a] - \beta \sum_{k\,\in\,\Lambda_a} [S_a I_k],$$

$$\frac{d\,[S_a S_b]}{dt} = -\beta \left(\sum_{k\in\Lambda_a} [I_k S_a S_b] + \sum_{k\,\in\,\Lambda_b} [S_a S_b I_k] \right) + \gamma\,([S_a I_b] + [I_a S_b]) \tag{5.3.6}$$

$$\frac{d\,[S_a I_b]}{dt} = -\,(\beta+\gamma)\,[S_a I_b] - \beta \left(\sum_{k\in\Lambda_a} [I_k S_a I_b] - \sum_{k\,\in\,\Lambda_b} [S_a S_b I_k] \right) + \gamma\,[I_a I_b]\,,$$

$$\frac{d\,[I_a I_b]}{dt} = \beta \left([S_a I_b] + [I_a S_b] + \sum_{k\in\Lambda_a} [I_k S_a I_b] + \sum_{k\in\Lambda_b} [I_a S_b I_k] \right) - 2\gamma\,[I_a I_b]\,.$$

Now, the system of differential equations (5.3.6) can be numerically solved by using the approximation given in equation (5.3.5). Note that our model differs from the one presented in [EK02a] and [Bau02] since we take into consideration the transitivity φ_{abc}, which turns out to have a significant effect on the outcome of the model.

Model (5.3.6) can be altered to obtain the system of differential equations (5.3.7), describing the dynamics of propagation in the SIR framework:

$$
\begin{aligned}
\frac{d\,[I_a]}{dt} &= \beta \sum_{k\in\Lambda_a} [S_a I_k] - \gamma\,[I_a]\,, \\
\frac{d\,[S_a]}{dt} &= -\beta \sum_{k\,\in\,\Lambda_a} [S_a I_k], \\
\frac{d\,[R_a]}{dt} &= \gamma\,[I_a]\,, \\
\frac{d\,[S_a S_b]}{dt} &= -\beta \left(\sum_{k\in\Lambda_a} [I_k S_a S_b] + \sum_{k\,\in\,\Lambda_b} [S_a S_b I_k] \right), \\
\frac{d\,[S_a I_b]}{dt} &= -\left(\beta+\gamma\right)[S_a I_b] - \beta \left(\sum_{k\in\Lambda_a} [I_k S_a I_b] - \sum_{k\,\in\,\Lambda_b} [S_a S_b I_k] \right), \\
\frac{d\,[S_a R_b]}{dt} &= -\beta \sum_{k\in\Lambda_a} [I_k S_a R_b] + \gamma\,[S_a I_b]\,, \\
\frac{d\,[I_a I_b]}{dt} &= \beta \left([S_a I_b] + [I_a S_b] + \sum_{k\in\Lambda_a} [I_k S_a I_b] + \sum_{k\in\Lambda_b} [I_a S_b I_k] \right) - 2\gamma\,[I_a I_b]\,, \\
\frac{d\,[I_a R_b]}{dt} &= \beta \sum_{k\in\Lambda_a} [I_k S_a R_b] + \gamma\left([I_a I_b] - [I_a R_b]\right).
\end{aligned}
\tag{5.3.7}
$$

5.3.1 Calculating R_0 for the SIR Model on a Scale-free Random Graph

Recent studies [PV02, BPV03] show that there is no epidemic threshold on scale-free graphs in large limit of their order, provided $2 < f < 3$. While this result entails that scale-free networks are tolerant to random damages, it also renders these networks a medium on which effective control of propagation would be difficult to achieve. Here, we show that finite uncorrelated scale-free graphs exhibit epidemic threshold—a function of the maximum degree. The implications of our result is the first analytical explanation that the removal of nodes of highest degrees can inhibit the propagation.

The basic reproductive ratio R_0 is defined as *the average number of secondary infectious nodes produced by an average infectious node in a totally susceptible population* [AM92]. When R_0 is greater than 1 a network worm (or a disease) can invade and increase within such population, whereas when R_0 is less than 1 any invasion is doomed to deterministic extinction (although stochastic effects can make a difference, especially close to the R_0 boundary). Hence, R_0 is an epidemic threshold—a

fundamental quantity in epidemiology and control of epidemics. In practice R_0 is calculated from the initial growth rate of an infinitesimal infection in an otherwise susceptible population [Kee99]. Thus, using the first equation in model (5.2.8), one may obtain that $R_0 = \dfrac{\beta \overline{d}}{\gamma}$.

Earnes and Keeling [EK02a] already observed that R_0 for model (5.3.7) is given by:

$$R_0 = \frac{\beta\,(\lambda - 1)}{\gamma}, \tag{5.3.8}$$

where λ is the dominant eigenvalue of the matrix C whose entries c_{ab} are given by:

$$c_{ab} = \frac{[ab]\,(b-1)}{b[b]}. \tag{5.3.9}$$

The matrix C is therefore a useful means of quantifying the connectedness of the graph on which propagation takes place.

Mihail and Papadimitriou [MP02] showed that the largest eigenvalues of a power law graph with exponent f has power law distribution if the exponent f of the power law graph satisfies $f > 3$, and, thus, verified the conjecture in Faloutsos *et al.* [FFF99]. Chung *et al.* [CLV03] studied the spectrum distribution of random graphs with given expected degrees, and proved that (under certain mild conditions) the eigenvalues of the (normalized) Laplacian of a random power-law graph follow the semicircle law,

whereas the spectrum of the adjacency matrix of a power-law graph obeys the power law.

In this section, we analyze the spectrum of matrix C defined by equation (5.3.9) in order to determine R_0 for a propagation on a given scale-free random graph. We start by stating two lemmas.

Lemma 5.3.1. *If $A_{n\times n}$ is nonsingular, and if c and d are $n \times 1$ vectors, then:*

$$\left|A + cd^T\right| = |A| \left(1 + d^T A^{-1} c\right).$$

Proof. Write $A + cd^T = A\left(I + A^{-1}cd^T\right)$. Observe that

$$\begin{pmatrix} I & 0 \\ d^T & 1 \end{pmatrix} \begin{pmatrix} I + A^{-1}cd^T & A^{-1}c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \\ -d^T & 1 \end{pmatrix} = \begin{pmatrix} I & c \\ 0 & 1 + d^T A^{-1} c \end{pmatrix}.$$

Simple application of product-rules for determinants yields the result. □

Lemma 5.3.2. *If $A_{n\times n}$ is of the form*

$$A = \begin{pmatrix} \frac{1+\alpha_1}{\alpha_1} & 1 & 1 & \cdots & 1 \\ 1 & \frac{1+\alpha_2}{\alpha_2} & 1 & \cdots & 1 \\ 1 & 1 & \frac{1+\alpha_3}{\alpha_3} & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & \frac{1+\alpha_n}{\alpha_n} \end{pmatrix},$$

then

$$|A| = \frac{1+\sum\limits_{i=1}^{n}\alpha_i}{\prod\limits_{i=1}^{n}\alpha_i}.$$

Proof. Write $A = D + ee^T$, where D is a diagonal $n \times n$ matrix with entries $d_{i,i} = \frac{1}{\alpha_i}$, and e is a one-vector. Direct application of Lemma 1 gives:

$$|A| = |D|\left|I + e^T D^{-1} e\right| = \frac{1+\sum\limits_{i=1}^{n}\alpha_i}{\prod\limits_{i=1}^{n}\alpha_i}.$$

□

Our main result is:

Theorem 5.3.3. *The reproduction ratio R_0 for epidemics (propagation) on an uncorrelated scale-free graph with n nodes and degree distribution $P(d(v) = k) \sim k^{-f}$*

and maximum degree Δ *is:*

$$R_0 = \frac{\beta(\lambda - 1)}{\gamma},$$

where

$$\lambda = \left| \frac{\Delta^{3-f}}{3-f} - \frac{\Delta^{2-f}}{2-f} - \frac{2^{3-f}}{3-f} + \frac{2^{2-f}}{2-f} \right|.$$

Proof. Given a scale-free degree distribution $P(d(v) = k) = k^{-f}$ of a graph G on n nodes, the expected number of nodes of degree a is $[a] = na^{-f}$, while the expected number of nodes of degree b is $[b] = nb^{-f}$. The number of edges incident on nodes of degree a and b could be calculated as $[ab] = n \sum_{k=1}^{n} k^{1-f} P(a, b)$, where $P(a, b)$ is the probability that an edge chosen uniformly at random from G is incident on nodes of degree a and b. By substituting these quantities, obtained from the scale-free graph G, in equation (5.3.9), one obtains the following expression for the elements of the matrix C:

$$c_{ab} = \frac{(b-1) P(a, b) n \sum_{k=1}^{n} k^{1-f}}{nb^{1-f}} = \overline{d}(b-1) b^{f-1} P(a, b). \tag{5.3.10}$$

In general, for a finite graph G with maximum degree Δ, C is a $\Delta \times \Delta$-matrix. We also note that the unlike the adjacency matrix of G, the modified contact matrix

C is not symmetrical, and has the following form:

$$C = \overline{d} \begin{pmatrix} 0 & 2^{f-1}P(1,2) & 2\cdot 3^{f-1}P(1,3) & \cdots & (i-1)\,i^{f-1}P(1,i) & \cdots \\ 0 & 2^{f-1}P(2,2) & 2\cdot 3^{f-1}P(2,3) & \cdots & (i-1)\,i^{f-1}P(2,i) & \cdots \\ 0 & 2^{f-1}P(3,2) & 2\cdot 3^{f-1}P(3,3) & \cdots & (i-1)\,i^{f-1}P(3,i) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 2^{f-1}P(i,2) & 2\cdot 3^{f-1}P(i,3) & \cdots & (i-1)\,i^{f-1}P(i,i) & \cdots \\ \vdots & \vdots & \vdots & & \vdots & \ddots \end{pmatrix} \tag{5.3.11}$$

The characteristic polynomial of C is of the form:

$$p_{\Delta}(\lambda) =$$

$$\overline{d}^{\Delta} \begin{vmatrix} -\lambda & 2^{f-1}P(1,2) & 2\cdot 3^{f-1}P(1,3) & \cdots & (i-1)\,i^{f-1}P(1,i) & \cdots \\ 0 & 2^{f-1}P(2,2)-\lambda & 2\cdot 3^{f-1}P(2,3) & \cdots & (i-1)\,i^{f-1}P(2,i) & \cdots \\ 0 & 2^{f-1}P(3,2) & 2\cdot 3^{f-1}P(3,3)-\lambda & \cdots & (i-1)\,i^{f-1}P(3,i) & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 2^{f-1}P(i,2) & 2\cdot 3^{f-1}P(i,3) & \cdots & (i-1)\,i^{f-1}P(i,i)-\lambda & \cdots \\ \vdots & \vdots & \vdots & & \vdots & \ddots \end{vmatrix} =$$

$$-\frac{\lambda\overline{d}^{\Delta}(\Delta!)^{f}}{\Delta} \begin{vmatrix} P(2,2)-\dfrac{\lambda}{2^{f-1}} & P(2,3) & \cdots & P(2,i) & \cdots \\ P(3,2) & P(3,3)-\dfrac{\lambda}{2\cdot 3^{f-1}} & \cdots & P(3,i) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ P(i,2) & P(i,3) & \cdots & P(i,i)-\dfrac{\lambda}{(i-1)\,i^{f-1}} & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{vmatrix}. \tag{5.3.12}$$

For scale-free graphs without degree-correlation, the joint probability distribution could be written, without loss of generality, as $P(a,b) = \dfrac{abP(a)\,P(b)}{\overline{d}^{\Delta}}$, where $P(a)$ stands for $P(a) = P(d(v)=a)$. The characteristic polynomial given in equation (5.3.12) obtains the following form:

$$p_{\Delta}(\lambda) = -\frac{\lambda (\Delta!)^{2-f}}{\Delta d^{\Delta}} \begin{vmatrix} 1-\frac{\lambda}{2^{1-f}} & 1 & \cdots & 1 & \cdots \\ 1 & 1-\frac{\lambda}{2\cdot 3^{1-f}} & \cdots & 1 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & 1 & \cdots & 1-\frac{\lambda}{(i-1)\, i^{1-f}} & \cdots \\ \vdots & \vdots & & \vdots & \ddots \end{vmatrix}. \tag{5.3.13}$$

Closer inspection of the matrix on the right-hand side of equation (5.3.13) shows that, for a fixed Δ, it has the same form as the matrix in Lemma 1. Using Lemma 2, it is clear that the characteristic polynomial can be further simplified to get:

$$\begin{aligned} p_{\Delta}(\lambda) &= \left(-\frac{\lambda}{\overline{d}}\right)^{\Delta} \frac{(\Delta!)^{2-f}}{\Delta} \frac{\left[1-\frac{1}{\lambda}\sum\limits_{k=2}^{\Delta}(k-1)\, k^{1-f}\right]}{\prod\limits_{k=2}^{\Delta}(k-1)\, k^{1-f}} = \\ &= (-1)^{\Delta}\left(\frac{\lambda}{\overline{d}}\right)^{\Delta}\left[1-\frac{1}{\lambda}\sum_{k=2}^{\Delta}(k-1)\, k^{1-f}\right] = 0. \end{aligned} \tag{5.3.14}$$

From equation (5.3.14), for a fixed Δ, the dominant eigenvalue is given by:

$$\lambda = \left|\int_{2}^{\Delta}(k-1)\, k^{1-f} dk\right|,$$

i. e.,

$$\lambda = \left| \frac{\Delta^{3-f}}{3-f} - \frac{\Delta^{2-f}}{2-f} - \frac{2^{3-f}}{3-f} + \frac{2^{2-f}}{2-f} \right|.$$

□

5.4 Pair-approximation Model vs. Individual-based Simulation

In this section, we test the accuracy of the pair-approximation model (5.3.6) by comparing its numerical solution to results from: (1) the individual-based simulation of the worm propagation on a Macroscopic Internet graph (on n nodes and average degree $\overline{d}$), and (2) the standard SIS model (which ignores correlation) on two topologies: the complete graph on n nodes (model (5.2.5)) and the Erdös-Renyi graph on n nodes and average degree $\overline{d}$ (model (5.2.4)).

The empirical study is conducted on *Macroscopic Internet graphs*, defined in Chapter 2, Section 2.2. To obtain the Macroscopic Internet graphs, we used the data for inter-connectedness of the Internet on the Autonomous System level collected by the University of Oregon Route View Project [Pro] and made available by NLANR (National Laboratory of Applied Network Research). We considered snapshots of the

Internet of various order and size, shown in Figure 5.1. The pre-processing step consists of determining parameters for model (5.3.6): for given degrees a, b, and c, the number of adjacent nodes of degree a and b, the set Λ_a, and the transitivity φ_{abc} are determined.

Date	Order	Size
08.11.1997	3015	5156
02.04.1998	3522	6324
03.07.1998	3797	6936
02.10.1998	4180	7768
14.01.1999	4517	8376
02.04.1999	4885	9276
02.07.1999	5357	10328
02.10.1999	5861	11313
02.01.2000	6474	12572
03.04.2000	7246	14629
02.07.2000	7956	15943
02.10.2000	8836	17823
02.01.2001	9048	18172
16.03.2001	10515	21455

Figure 5.1: Macroscopic Internet graphs used in simulations

Next, we developed an individual-based simulation of the stochastic propagation process on a Macroscopic Internet graph. The individual-based simulation has two advantages: First, the propagation process and the underlying topology can be controlled to simulate different scenarios. Second, this simulation provides very precise

and detailed information about the propagation dynamics without biases which might be present in real data [Moo03]. The individual-based simulation combines Monte Carlo simulation of events, taking place at given rates, with an event-scheduler that determines the order in which events happen on a given graph (medium for propagation) and, thereby, affect the states of the nodes. The scheduler is implemented as a priority queue. There are two types of events that can take place in the Susceptible-Infectious-Susceptible model: infection and curing. If a node u is infectious, it attempts infection of each of its neighbors at rate β. There is also the event that node u is cured, and, thus, become susceptible, at rate γ. Let node u be cured at time t, and be re-infected at time $t + dt$. Any infection generated by the node u in the time interval $[t, t + dt)$ is discarded by the scheduler. The event of node u attempting infection of an already infectious neighbor v at time t is also discarded by the scheduler.

We simulated worm propagation in the Susceptible-Infectious framework in order to determine the time the worm takes to infect all nodes of a given graph G. Simulations were performed on ten Macroscopic Internet graphs (the results for five graphs, identified by the top entry in the left-most column appear in Figure 5.2). To determine how choice of the initial node influences the propagation, we first determined the labels of three nodes with smallest degrees and three nodes with highest degree, shown in the first and second column of each table in Figure 5.2. The rest of the entries show the average time over 100 simulations for the worm to propagate on all

nodes by starting from a pre-specified initial node and spreading with infectious rate β. The general results of the experiments can be summarized as follows:

1. *The time to propagate to all nodes decreases with the increase of the degree of the initial node.*

   As an infectious node of higher degree has bigger pool of susceptible nodes, it gives the worm the possibility to establish a considerable fraction of infectious nodes in the early stages of the propagation. On average, the propagation to all nodes of G initiated from a node of maximum degree takes time by 5% shorter compared to the propagation that starts from a node of minimum degree.

2. *The time to propagate to all nodes increases with the increase of the order of the graph.*

   As discussed in Chapter 2, the diameter of two graphs G_1 and G_2, $|V(G_1)| < |V(G_2)|$, whose degree distributions follow the same scale-free distribution, are such that $D(G_1) < D(G_2)$. Therefore, on a graph with greater diameter the worm takes longer time to infect all nodes.

3. *The time to propagate to all nodes does not strictly decreases with the increase of the infectious rate β.*

   In other words, there is a value of the infectious rate β at which the function $t(\beta)$ has a local minimum, as shown in Figure 5.3. According to the simulation results shown in Figure 5.2, the value of $\beta = 1.5$ seems to be invariant and depends only on the exponent of the scale-free degree distribution of the graph G. The reason for such behavior is that, at the local minimum, rapidly-building correlations between the states of adjacent nodes hinder the propagation by lowering the number of available susceptible nodes. This observation is of *particular interest* as it provide the means to "control" the propagation of a fast-spreading worm by reducing its rate to the threshold value.

Figures 5.4 and 5.5 below, show the number of infectious nodes as a function of time, comparing the three deterministic models with two results of the stochastic individual-based simulation. Neither the mean-field model nor the Erdös-Renyi (or a $\overline{d}$-regular graph on n nodes) satisfactorily predicts the level of propagation (*i.e.*, the number of infectious nodes at a given time). The second-order Runge-Kutta numerical solution of the proposed pair-approximation model (5.3.5), (5.3.6) with results of the second pre-processing task as input, performs matches the results of the individual based simulation.

The model on $\overline{d}$-regular graphs underestimates the equilibrium level because it does not include the nodes of high degrees (*i.e.*, the *core* of the scale-free graph). The mean-field model consistently over-estimates the number of infectious nodes because the correlations and graph structure, that may inhibit propagation, are ignored. In contrast, our pair-approximation model includes both nodes of various degrees and correlations between states of nodes, and gives an accurate representation of the stochastic propagation process.

Remark 5.4.1. Two algorithms were compared to numerically solve the system of ordinary differential equations with boundary conditions—Euler's method and Runge-Kutta method. In both cases, due to the complexness of the system, for small values of the time change dt we saw emergence of attractors. For small values of dt, such as 0.001 used in the simulation, even the Euler's method produces good results.

With the help of the individual-based simulation one can also study the distribution of infectious nodes based on degrees. A typical example of this distribution is shown in Figure 5.6. The results show that the number of infectious nodes rapidly increases when a node of higher degree is infected.

5.5 Summary

Developing an accurate model for the worm propagation is of critical importance not only for understanding better the worm's behavior but also for devising techniques to contain such cyber attacks. The existing studies include either simulation of worm propagation on a macroscopic level or a system of differential equation for propagation on Erdös-Renyi and regular graphs, as pointed in the survey. Moreover, in all existing models of worm propagation, it is not evident how the network structure might affect the dynamics of the stochastic propagation process. Our contribution here was twofold: (1) a model of propagation on a scale-free graph G which takes as input the number of nodes, number of edges, number of edges incident on nodes of certain degrees, and transitivity of the graph G, and (2) implementation of an individual-based simulation for worm propagation that can be cast in different epidemiological frameworks. The accuracy of the model was tested by comparing the numerical solution of the pair-approximation model to the results from the individual-based simulation on scale-free Macroscopic Internet graphs. The results show excellent agreement between the results from the model and the individual-based simulation. Due to its accuracy, this model (incorporating graph-theoretic invariants) has a great potential to be used in developing realistic techniques for propagation control.

AS graph 08.11.1997		beta				
		0.2	0.5	0.9	1.5	1.8
min degree	node					
1	14	18.0693	7.77851	4.62873	3.11892	3.48948
2	13	17.766	7.68768	4.60186	3.12606	3.33325
3	15	17.5662	7.47541	4.61656	3.13278	3.30167
max degree	node					
590	4	16.4317	7.08121	4.28161	2.95023	3.22838
524	7	16.1565	6.90071	4.35007	2.87707	3.30877
355	6	16.4247	7.27244	4.46792	3.01278	3.26648

AS graph 02.10.1998		beta				
		0.2	0.5	0.9	1.5	1.8
min degree	node					
1	47	18.3675	7.89627	4.74167	3.20504	3.55996
2	17	18.0627	7.91097	4.6643	3.19542	3.46717
3	22	18.4302	7.76855	4.69197	3.14913	3.33457
max degree	node					
590	5	17.158	7.73703	4.45511	3.01981	3.34582
524	12	16.9676	7.37089	4.50048	2.96612	3.39495
355	10	17.2982	7.48106	4.5871	3.10994	3.2987

AS graph 02.07.1999		beta				
		0.2	0.5	0.9	1.5	1.8
min degree	node					
1	63	18.3734	7.98559	4.81368	3.24062	3.42445
2	19	18.5593	8.17793	4.70665	3.19895	3.48956
3	15	18.5207	7.93869	4.74685	3.18794	3.343
max degree	node					
1193	2	17.4871	7.43152	4.44526	2.98155	3.39874
674	10	17.6827	7.58919	4.57659	3.11	3.41901
588	7	17.5386	7.70924	4.63347	3.10909	3.38447

AS graph 02.07.2000		beta				
min degree	node	0.2	0.5	0.9	1.5	1.8
1	15	19.8676	8.42701	5.11502	3.42114	3.71483
2	18	20.1391	8.62913	5.15767	3.41545	3.56204
3	23	19.4788	8.44671	5.05909	3.3409	3.5603
max degree	node					
1772	2	18.9615	8.0676	4.97258	3.29313	3.48871
961	9	19.2357	8.28964	4.90184	3.33247	3.49154
802	7	18.9133	8.24647	4.96538	3.34996	3.47815

AS graph 16.03.2001		beta				
min degree	node	0.2	0.5	0.9	1.5	1.8
1	44	21.0673	8.91663	5.35035	3.57629	3.76316
2	37	21.385	8.90863	5.35318	3.47916	3.74718
3	34	20.6299	8.93965	5.33861	3.54236	3.64725
max degree	node					
2277	2	20.1728	8.44475	4.98278	3.37857	3.58247
1231	13	20.3132	8.67817	5.248	3.46303	3.62322
899	15	20.4644	8.87768	5.2025	3.42517	3.69419

Figure 5.2: Time to propagate to all nodes of a Macroscopic Internet graph for five different values of the parameter β.

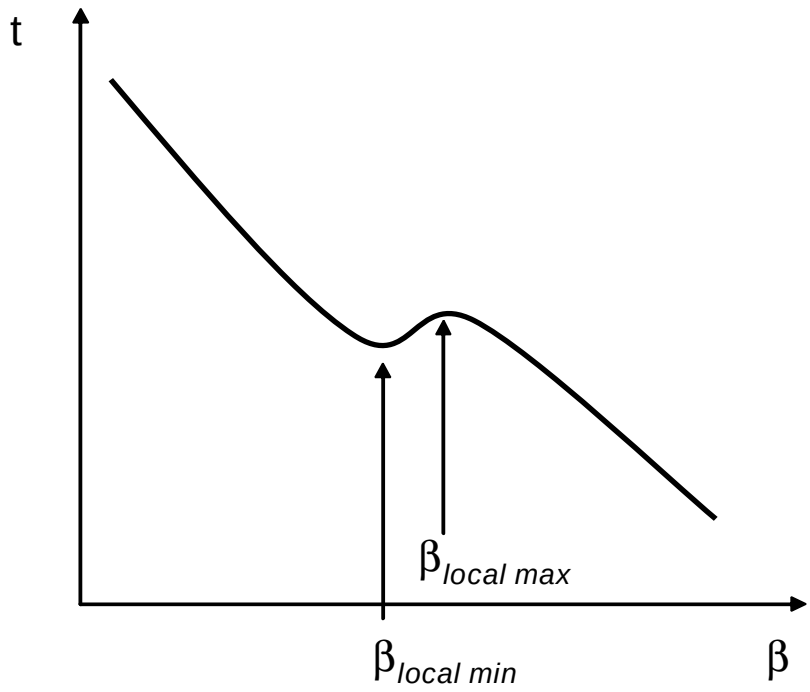


Figure 5.3: Time to infect all nodes as a function of the rate β is not a strictly decreasing function

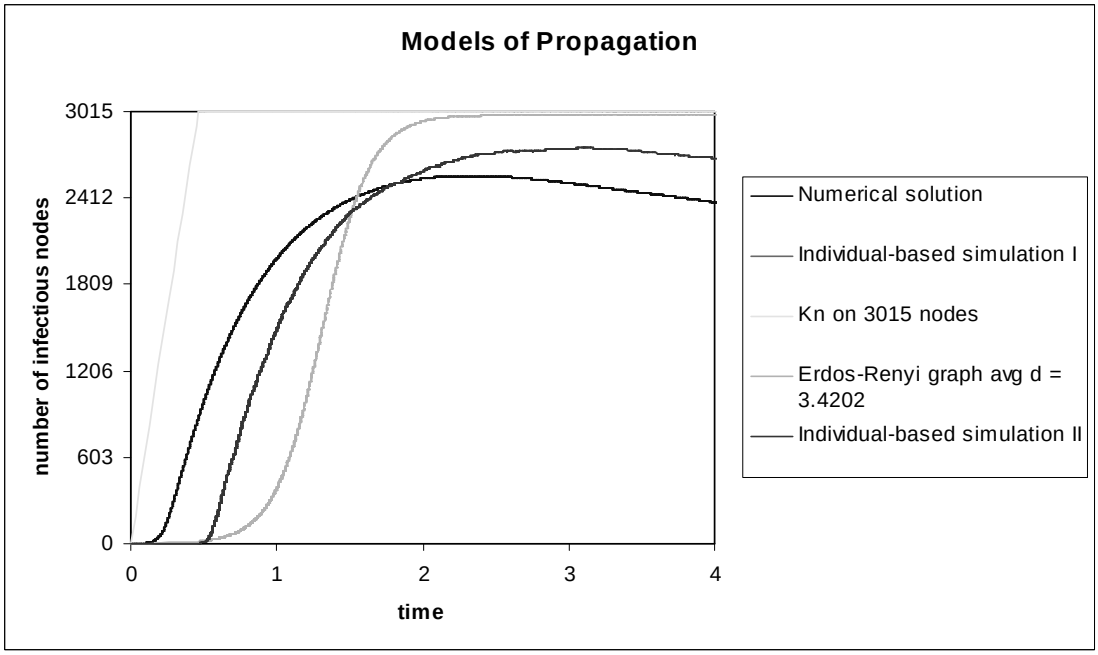


Figure 5.4: Susceptible-Infectious-Susceptible models of propagation—numerical solution of pair-approximation model, individual-based simulation of propagation on an Internet graph $n = 3015$ and $m = 5156$, propagation on complete graph $n = 3015$, propagation on Erdos-Renyi random graphs with $\overline{d} = 3.4202$; parameters of propagation $\beta = 1.8, \gamma = 0.05$

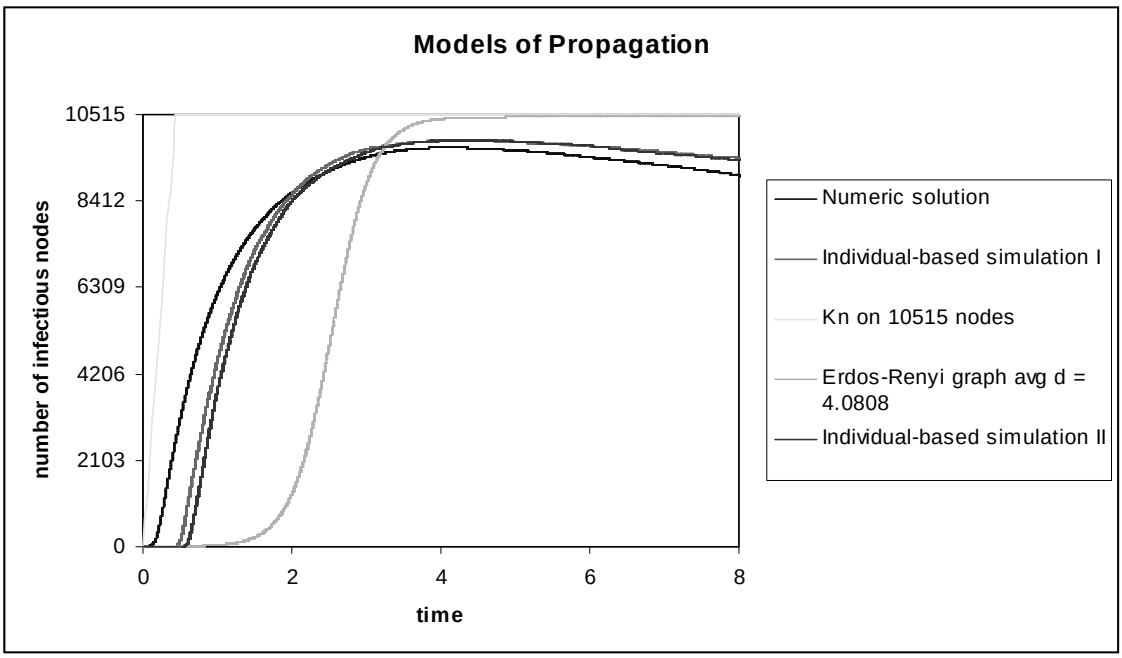
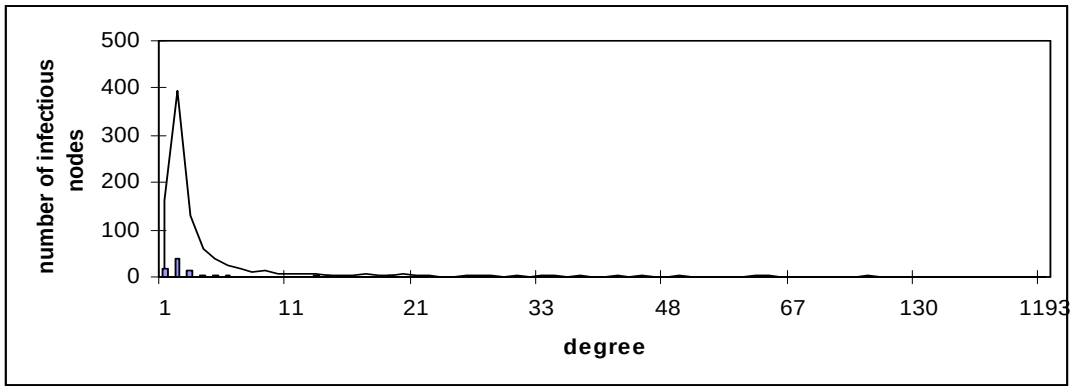
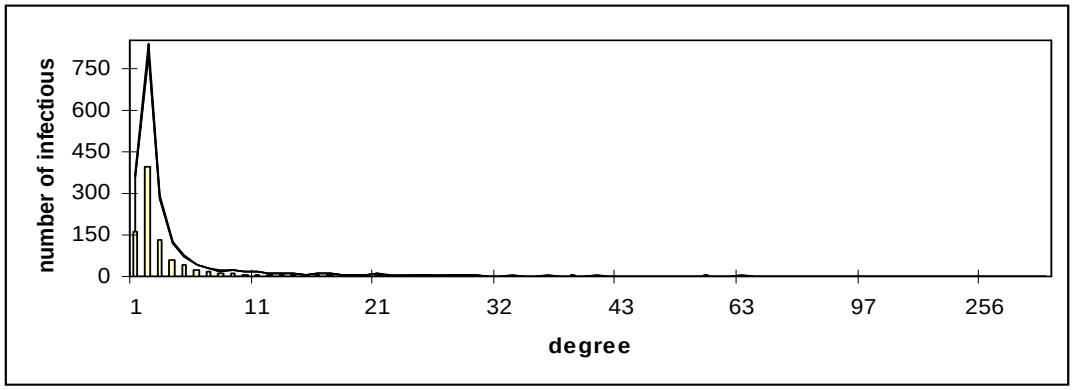


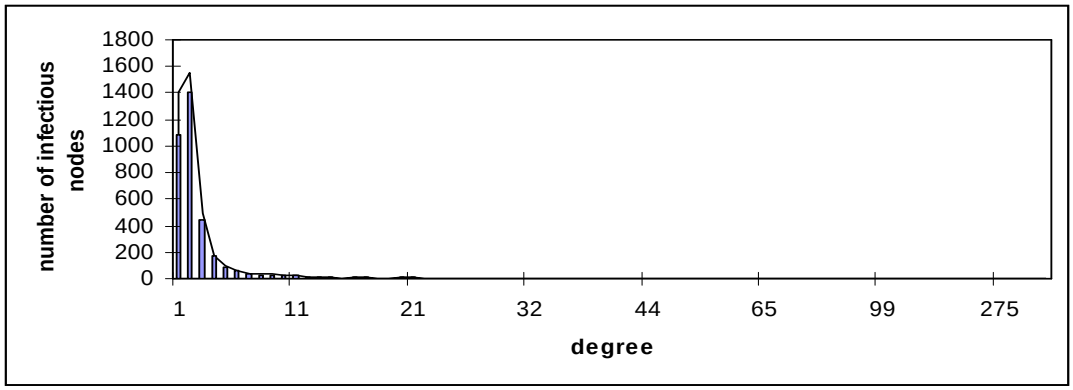
Figure 5.5: Susceptible-Infectious-Susceptible models of propagation—numerical solution of pair-approximation model, individual-based simulation of propagation on an Internet graph $n = 10515$ and $m = 21455$, propagation on complete graph $n = 10515$, propagation on Erdos-Renyi random graphs with $\overline{d} = 4.0808$; parameters of propagation $\beta = 0.9, \gamma = 0.02$


number of infectious nodes
500
400
300
200
100
0
1
11
21
33
48
67
130
1193
degree

(a)


number of infectious
750
600
450
300
150
0
1
11
21
32
43
63
97
256
degree

(b)



(c)

Figure 5.6: Distribution of number of infectious nodes pre degree on Macroscopic Internet graph from 02.07.1999; $\beta = 1.5, \gamma = 0.1$ (t_1 shown with bars, t_2 with line) (a)time moments, $t_1 = 1.00536$ and $t_2 = 0.28634$, (b) time moments, $t_1 = 0.28634$ and $t_2 = 0.42069$, (c)time moments, $t_1 = 0.932814$ and $t_2 = 0.47181$

CHAPTER 6

CONTROL STRATEGIES ON SCALE-FREE GRAPHS

6.1 Introduction

Despite the recent surge of research in control of worm propagation, currently, there is no effective defense system against such cyber attacks. The existing automated network-security solutions (*e.g.*, anti-virus software, firewalls, and intrusion detection systems) and human-dependent counter-measures (*e.g.*, software-patching, traffic-blocking) have been deemed inadequate for effective control of worms [Ins04, WPS03a, WPS03b]. Therefore, devising new control strategies is a first step towards a comprehensive network-security solution.

Recent studies [NL04] of control strategies have found the topology on which the worm propagates, from a node to its neighbors, to be significantly different from the underlining network infrastructure (*e.g.*, the Internet). This is true only for network

worms that scan the IP space uniformly at random, and therefore can communicate with any host on the network. Although the introduction of the Internet has arguably made the assumption of sparseness (of the propagation topology) no longer valid, the idea of *locality*, especially in the case of analytical modeling of localized propagation strategies (see Chapter 3) is still applicable, and, therefore, used in developing novel control strategies.

Our contribution here is twofold: (1) a classification of existing control strategies (with a discussion of their advantages and disadvantages). As propagation and control strategies are tightly coupled with a particular detection mechanism, the review of control strategies also encompasses the existing detection mechanism, and (2) analysis of *five* control strategies. Like in Chapter 5, we use the epidemiological approach with information about the network structure.

6.2 Determinants of Propagation and Control

Due to the strong analogy between network worms and infectious diseases, epidemiological models have been widely used in modeling not only worm's propagation, but also detection and control strategies. Traditional epidemiology has identified three factors determining the outcome of an infection [Het00]: the size of the susceptible population, the length of the infectious period, and the rate of infections. Like with

diseases, there are two potential approaches to mitigate cyber attacks with network worms: *prevention*, that includes technologies for reducing the size of the susceptible population, and *control*, that consists of strategies for reducing any of the three factors determining the outcome of the propagation.

As system design and implementation is prone to human (logic) errors, we believe that any prevention technique, by itself, cannot suffice in countering network worms. Control strategies, implemented in existing anti-virus solutions, are capable of: (1) reducing the size of the susceptible population (by *immunizing*, *e.g.*, patching a portion of the susceptible), (2) shrinking the infectious period (by *eliminating the worm copies*), and (3) limiting the infection rate (by *disabling communications*). The time from detection of a vulnerability to the design, implementation, and testing of its patch is long (expressed in terms of days). Moreover, as the process of deploying patches, in many cases, is not automated, this strategy, although essential, cannot provide timely control. Control strategies that lower the infectious period require the *worm signature*—string of bytes in the traffic that pass through a network link—to be known. In absence of a patch or worm signature, *quarantining mechanisms* can prevent the worm from propagating by disabling communication directed from a host, suspected to be or detected as, compromised.

There are three quarantining mechanisms that act towards reducing the size of the susceptible population or limiting the infection rate by disabling communication:

(1) content-filtering, (2) address–blacklisting, and (3) traffic-blocking (from a subset of ports). In content-filtering, which requires a database of worm signatures, the control system drops *only* the packets containing one of these signatures. In address-blacklisting, which involves a list of hosts that have been suspected or identified as compromised, the control system drops *all* packets coming from these hosts. In the third alternative, the responders of the control system disable *all* or *a portion* of the traffic (from a subset of ports) coming from a host that has been suspected or identified as compromised. If traffic-blocking is applied to all ports, we will say that the host has been *disconnected*.

6.3 Classification of Control Strategies

Implementation of control strategies involves deployment of an *agent-based control system* composed of sensors, aggregators, and responders. *Sensors* are programs designed for detecting an anomaly (in network traffic or host behavior), indicating that worm propagation has started. *Aggregators* communicate with the sensors to gather information about the global characteristics of the propagation and plan further actions, *e.g.*, alarming certain responders. Finally, *responders*, through two-way communication with aggregators and sensors, initiate a pre-specified control strategy.

The responders may be implemented as a non-replicating agent or as a self-replicating agent (called *good worm*).

Although the deployment of an agent-based control system imposes many practical challenges, such as: design of communication protocol specific to the system, integration with the existing intrusion-detection systems, and insuring the system's robustness to attack, the effectiveness of a control strategy could still be analyzed by means of models and simulation.

Remark 6.3.1. We distinguish between *control mechanisms* and *control strategies*. A *control strategy* specifies when and which of the responders are activated. A *control mechanism* specifies how the responders act towards hindering the propagation, *i.e.*, it specifies how a given control strategy is implemented. We will discuss two types of control strategies—quarantining and immunization.

Worms propagate via network communications in a similar way as a virus spreads among people. Since network communications can be modeled by a graph, in which nodes represent hosts and edges are communication lines, we will use graph–theoretic terms to describe the existing epidemiological models of control: The propagation takes place on a graph $G = (V, E)$ with n nodes and m edges. Let t_d denote the time when the control system has detected the worm, t_r refer to the *reaction time* of the system, *i.e.*, the time needed for *all* responders to activate the control mechanism, and t_q denote the time during which a control strategy is applied on susceptible nodes.

Depending on whether all responders of the control system become active at once or only a fraction of responders is activated to counter the propagation, control strategies can be *static* or *dynamic*. In *static quarantining strategies*, all responders are activated at once, at time moment $t_d + t_r$. More specifically, for address–blacklisting (respectively, traffic-blocking), if a node u is detected as infectious at time $t_d(u)$, the quarantining system drops all packets (respectively, packets from a subset of ports) coming from node u after time $t_d(u) + t_r$. Static quarantining strategies require global distribution of information about the worm presence. Hence, they may impose unnecessary disruption of normal network functions, as the propagation may be countered when only a portion of the responders are active. In *dynamic quarantining*, the quarantining system makes a decision about how many and which responders to be activated based on the local information about the propagation of the worm. While in *static immunization*, the given portion of the susceptible nodes is immunized (*i.e.*, removed) at only one time moment, in *dynamic immunization*, the decision of which and how many nodes should be immunized depends on the dynamics of the propagation. Clearly, dynamic control strategies are time-dependent and may change based on the amount and quality of available information.

Let $S(t)$ denote the number of susceptible nodes at time t, $Q_s(t)$ the number of quarantined-susceptible nodes, $R_s(t)$ the number of removed-susceptible nodes, $I(t)$ the number of infectious nodes, $I_d(t)$ the number of detected infectious nodes,

$I_u(t)$ the number of undetected infectious nodes, $Q(t)$ the number of quarantined-infectious, and $R(t)$ denote the number of removed nodes. Let β denote the rate with which an infectious node infects its adjacent susceptible nodes.

First, we describe the simplest epidemiological model—the Susceptible-Infectious—that can be used in the study of the *worst-case propagation*, *i.e.*, when control strategies are not available. In this class of models, once a susceptible node becomes infectious, it does not change its state. Let the average degree of an infectious node be $\overline{d}$, and the fraction of infectious nodes at time t be $i(t)$. The expected number of susceptible neighbors that can be infected by a given infectious node is $\overline{d}(1 - i(t))$. Since there are $I(t)$ infectious nodes, the total rate of newly-infected nodes is $\beta\overline{d}(1 - i(t))\, i(t)$. This model is described by the differential equation (6.3.1):

$$\frac{di(t)}{dt} = \beta\overline{d}(1 - i(t))\, i(t)\,, \tag{6.3.1}$$

with boundary conditions: $i(0) = \dfrac{I(0)}{n} > 0$ and for all $t \geq 0$, $i(t) + s(t) = 1$. The solution of equation (6.3.1) for the fraction of infectious nodes is the *logistic curve*: $i(t) = \dfrac{i(0)\ e^{\beta' t}}{1 - i(0) + i(0)\ e^{\beta' t}}$, where $\beta' = \beta\overline{d}$.

6.3.1 Models of Static Control Strategies

Static control strategies change the structure of the graph on which the worm propagates. Let V_{I_u} be the set of undetected infectious nodes, V_{I_d} the set of detected infectious nodes, and V_{R_s} be the set of removed susceptible nodes. From time $t_d + t_r$ onward, (1) in the static content-filtering the propagation of the worm is from infectious nodes in $V_{I_u} \cup V_{I_d}$ on the graph $G - V_{R_s}$, (2) in address-blacklisting, the propagation of the worm is from infectious nodes in V_{I_d} on the graph $G - V_{R_s}$, while the worm continues propagating as unconstrained on the graph G from infectious nodes in V_{I_u}, and (3) in traffic-blocking, there is only propagation from infectious nodes in V_{I_u} on the graph $G - V_{R_s}$. Note that the conditions of traffic-blocking are equivalent to those of static immunization.

6.3.1.1 Static Immunization

Pastor-Satorras *et al.* [PV02] conducted simulation study to investigate the effects of immunizing nodes before the start of propagation. If V_{R_s} is the set of removed (immunized) nodes, the worm propagates on the graph $G - V_{R_s}$. The study demonstrated that static immunization, when V_{R_s} consists of nodes chosen at random, is inefficient in slowing down the propagation; however, when V_{R_s} consists of nodes of

highest degrees, static immunization can significantly inhibit the propagation. While the latter result seems interesting, the authors argue that detecting nodes of high degrees in scale-free networks is a difficult problem.

Similarly, the simulation study of Wang *et al.* [WKE00] examines the effects of immunization on two topologies: rooted trees and clustered networks (composed of cliques inter-connected with small number of edges). The parameter of their simulation is the *propagation fan out*—number of nodes to which the worm can send replicas at each time step. The time needed for the worm to propagate from one node to another is assumed to be one time tick. The first set of simulation is conducted on networks without immunization to determine the number of times a node is re–infected (called *re-infection count*). Two types of immunization are simulated—random and selective. Random immunization performs better on rooted trees as there is only one path between any two nodes; thus, it is possible to cut off an entire sub-tree of the network, which is not the case with the clustered network. For the case selective traffic-blocking in rooted trees, V_{R_s} is composed of nodes with highest re-infection counts (note, these nodes coincide with nodes with largest degrees). In the case of clustered networks, two strategies are used: in the first, nodes in V_{R_s} are chosen based on the re-infection count, while in the second, nodes of V_{R_s} from among those with highest weighted sum of the inter–cluster and inner-cluster degrees. The first strategy was able to contain the propagation, but results in a higher propagation

rate. The second could slow down the propagation rate, but was unable to contain the propagation.

The principal disadvantage of the studies in [WKE00] and [PV02] is that a fraction of nodes is removed before the worm starts propagating—an unrealistic assumption in itself. Another interesting open question is the effects of the longer propagation, resulting from the static immunization, on the functionality of the network.

6.3.1.2 Static Traffic-blocking

To overcome the problem of static immunization, Williamson *et al.* [WL03] analyzed a modification of the Susceptible-Infectious-Detected-Removed (SIDR) model. The aim of the study was to determine the effectiveness of the traffic-blocking approach called virus throttling [Wil02]. Virus throttling is an automatic mechanism for slowing a worm's propagation by limiting the rate of traffic. The model incorporates virus throttling by dividing the nodes into two groups—throttled and un-throttled. Here, an un-throttled node can be in one of the four states: susceptible, infectious, detected (in which the virus has been detected and cannot actively spread), and removed. Let p be the portion of throttled nodes. The model assumes complete graph as underlying topology, and involves two stages: in the first stage, prior to the release of the virus signature, nodes progress from susceptible to infectious state at rate β, according to

the model (6.3.1). In the second stage, after some time t_d when the worm is detected (*i.e.*, the worm signature is available), the throttled nodes are removed, so only the un-throttled infectious nodes (whose fraction is $i_u(t)$) can spread the worm:

$$\begin{aligned}
\frac{di_u(t)}{dt} &= \beta\overline{d}(1-p)\,s(t)\,i(t)\,, \\
\frac{di(t)}{dt} &= \beta\overline{d}s(t)\,i(t)\,, \quad t < t_d, \\
\frac{di(t)}{dt} &= \frac{di_u(t)}{dt} = \beta\overline{d}(1-p)\,s(t)\,i_u(t)\,, \quad t \geq t_d.
\end{aligned} \tag{6.3.2}$$

The study shows that when more than half of the nodes are throttled, even a late signature (*i.e.*, when t_d is relatively large) will result in a small outbreak.

It is clear that the effects of the traffic-blocking are determined by its deployment (*i.e.*, the choice of nodes on which it is applied). If p is the portion of nodes that have traffic-blocking mechanism, the number of infectious nodes with traffic-blocking mechanism is $pI(t)$, and the probability with which they infect susceptible nodes is $\beta' < \beta$. Thus, the number of infectious nodes can be described by the Susceptible–Infectious model given by:

$$\frac{di\,(t)}{dt} = [\beta' p + \beta\,(1-p)]\,\overline{d}\,(1 - i\,(t))\,i\,(t)\,, \tag{6.3.3}$$

with initial conditions same as for the model (6.3.1). Wong *et al.* [WWS04] performed a simulation study of model (6.3.2) on a complete graph, and showed that traffic-blocking at the backbone routers is effective. Traffic-blocking at edge routers is helpful for worm using random propagation strategy, but does little to suppress worms using local propagation strategies. Moreover, individual host-based traffic-blocking results in slight linear slowdown of the worm, regardless of the propagation strategy. The study used 23-day traffic traces from an edge router in the period of the Blaster worm.

To avoid installation of traffic-blocking on individual host, Chen and Tang [CT04] proposed a distributed anti-worm architecture (DAW) deployed at edge routers. Here, the agents (responders) use temporal and spatial rate-limiting algorithms based on the connection-failure rate: the temporal rate-limiting insures that packets are dropped from a host that has exceeded the normal connection-failure rate, while the spatial rate-limiting insures that packets are dropped from a sub-network that has exceeded its normal connection rate. Their simulation study shows that that temporal rate-limiting can slow down the propagation from minutes to days, while the spatial rate-limiting limits the number of infectious nodes to only 5% of the entire susceptible

population (the number of sub-networks is 10,000 with average of 10 susceptible per sub-network).

6.3.1.3 Static Content-filtering and Address-blacklisting

Moore *et al.* [MVS03] conduct simulations to analyze the effects of static content-filtering and address-blacklisting. The study is focused on three factors: reaction time, t_r, containment strategy, and deployment scenarios (not universal, but at some pre-selected nodes). The filter is placed on the shortest path between two nodes in the macroscopic Internet graph. Here, an approach is considered successful if it limits infection to 1% of the nodes within the 24 hour period. The conclusion of the simulation is that if the containment system is unable to activate filtering mechanisms within minutes of the start of propagation, the system will not be effective (with content-filtering performing better than address-blacklisting). Both approaches have the same weakness—it is unrealistic to think of a global blacklisting or content-filtering engine.

Compartmental epidemiological models are used for stratified population. The topology in these models is the *macroscopic Internet graph*, where every node represents a dense region—Autonomous System (AS). These models can be used to study intra-AS propagation, with the assumption that within an AS (with n_j nodes) the

worm propagates as on a complete graph K_{n_j}. The infectious attempts can then be modeled as being external or internal to an AS. If the macroscopic Internet graph has n nodes, the SI compartmental model can be written as:

$$\frac{di_j(t)}{dt} = \left[\sum_{k=1}^{n} \beta \frac{n_k}{N} i_k(t)\right] (1 - i_j(t)), \tag{6.3.4}$$

where, $1 \leq j \leq n$ and the parameter N denotes the total number of IP addresses. Serazzi *et al.* [SZ04] used model (6.3.4) to derive equations for the bandwidth consumption at each node. Nicole and Liljenstam [NL04] performed a study of static content-filtering and address-blacklisting, with results identical to those in Moore *et al.* [MVS03]. For the SIR compartmental model, Liljenstam *et al.* [LNB03] obtained:

$$\begin{aligned} \frac{di_j(t)}{dt} &= \left[\sum_{k=1}^{n} \beta \frac{n_k}{N} i_k(t)\right] (1 - i_j(t)), \\ \frac{dr_j(t)}{dt} &= \gamma i_j(t), \end{aligned} \tag{6.3.5}$$

where, $1 \leq j \leq n$. With model (6.3.5), Liljenstam *et al.* [LNB03] studied the destabilizing effects of worm propagation on the network infrastructure, since the compartmental approach allows for inclusion of limited details about communication protocols. In this simulation study, the scan traffic is model by using a combination of the

average scan rate, individual infection rates, and size of address space for each AS. This framework is further used to devise and study a detection mechanism based on ICMP unreachable messages (called ICMP-T3 messages) [BGB02], [BB03]. Simulations show that Code Red could have been detected by using ICMP–T3 messages when it had infected 0.2% of the susceptible nodes, while monitoring only 2^{17} *inactive* nodes (*i.e.*, two Class B networks that *are not assigned*); on the other hand, with 2^{18} monitored nodes (*i.e.*, four Class B networks), Slammer could have been detected when only 0.01% of the susceptible nodes were infected. Chen and Ranka [CR04] proposed a similar architecture in which ICMP–T3 messages are combined with TCP RESET packets to detect a worm employing random propagation strategy. However, in their approach, active nodes are also monitored.

6.3.2 Models of Dynamic Control Strategies

Dynamic control strategies are time-dependent and may change based on the amount and quality of available information. Variations of the traditional Susceptible-Infectious and Susceptible-Infectious-Removed models have been used to model several dynamic control strategies, reviewed in the next sub-sections.

6.3.2.1 Susceptible-Infectious Models of Dynamic Control Strategies

Nicole and Liljenstam [LN04] studied three control strategies carried out by self-replicating responders which spread at approximately the same rate as the worm. The problem of having a propagating worm and self-replicating agents (*i.e.*, good worm) in a susceptible population can be analyzed by the *Susceptible-Infectious models of two competing diseases*. Here, a node can be "infected" by either a worm or a self-replicating agent. Let $q_s(t)$ be the fraction of susceptible nodes infected (and, thus, quarantined) by the agent. If the self-replicating agent is endowed with a patch and the ability to discern only susceptible nodes, the model of dynamic immunization, on a graph with average degree $\overline{d}$, can be described as follows:

$$\begin{aligned} \frac{ds(t)}{dt} &= -\beta\overline{d}s(t)\left(i(t)+q_s(t)\right), \\ \frac{di(t)}{dt} &= \beta\overline{d}s(t)\,i(t), \\ \frac{dq_s(t)}{dt} &= \beta\overline{d}s(t)\,q_s(t), \end{aligned} \tag{6.3.6}$$

with boundary conditions $s(0) = \dfrac{n - I(0) - Q_s(0)}{n} \geq 0$, $i(0) = \dfrac{I(0)}{n} \geq 0$, $q_s(0) = \dfrac{Q_s(0)}{n} \geq 0$, $i(t) + q_s(t) + s(t) = 1$.

A self-replicating agent that can also detect an infectious node and blocks its traffic imposes interaction with the worm. Note that this control strategy combines dynamic immunization with dynamic traffic-blocking. The increased capabilities of the agent can be described by model (6.3.6) in which the second equation is changed as:

$$\frac{di\left(t\right)}{dt} = \beta\overline{d}\left(s\left(t\right) - q_s\left(t\right)\right) i\left(t\right). \tag{6.3.7}$$

Finally, the authors assume that the agent, which can also eliminate infectious nodes, spreads at greater rate than the worm, formulated as:

$$\frac{di\left(t\right)}{dt} = \beta\overline{d}s\left(t\right) i\left(t\right) - k\beta\overline{d}i\left(t\right) q_s\left(t\right), \tag{6.3.8}$$

where $k > 1$.

At any time moment t, the rate of propagation is proportional to the total number of infectious nodes $i\left(t\right) + q_s\left(t\right)$. Since, both types of infectious nodes send a great number of scans, these quarantining mechanisms may impair the functionality of the network with the increased traffic. Analysis of model (6.3.7) showed that the peak rate

of propagation is at least one third of the initial susceptible population—magnitude undesirable for real networks.

Furthermore, Nicole and Liljenstam [NL04] compared static and dynamic mechanisms. The dynamic mechanism that employs a patch is as good as the content-filtering provided some boundary conditions for the fraction of nodes included in the content-filtering infrastructure, while static content-filtering installed on the 30 most connected Autonomous Systems can outperform the dynamic mechanisms modeled by (6.3.6) – (6.3.8).

If a patch becomes available at time t_d and it is distributed at rate γ, the dynamics of immunization can be modeled by:

$$\begin{aligned}
\frac{di\,(t)}{dt} &= \beta\bar{d}s\,(t)\,i\,(t)\,, \quad t < t_d, \\
\frac{di\,(t)}{dt} &= \beta\bar{d}s\,(t)\,i\,(t) - \gamma i\,(t)\,, \quad t \geq t_d, \\
\frac{ds\,(t)}{dt} &= -\gamma s\,(t)\,.
\end{aligned} \tag{6.3.9}$$

Wong *et al.* [WWS04] studied the case when dynamic patching is combined with static traffic-blocking deployed on a portion of p nodes in a complete graph:

$$\frac{di(t)}{dt} = \beta s(t) I(t), \quad t < t_d,$$

$$\frac{di(t)}{dt} = [\beta' p + \beta(1-p)] s(t) I(t) - \gamma i(t), \quad t \geq t_d, \tag{6.3.10}$$

$$\frac{ds(t)}{dt} = -\gamma s(t).$$

The simulation of the model on a 1000-node scale-free graph shows that 80% of nodes are infected at the end of simulation without static traffic blocking, while when 20% are blocked (removed), 72% of the nodes become infectious.

In the field of worm detection, Zou *et al.* [ZGT03a] used model (6.3.1) on a complete graph to analyze a *trend-detection* mechanism based on the traffic-anomaly created by worms. The detection system is composed of distributed ingress and egress sensors for worm activity: when the monitoring system receives a surge of illegitimate scans, a Kalman filter is activated to estimate the parameter β. Since in the early stage the propagation exhibits exponential growth with constant, positive rate, the model can be described by $I(t) = (1 + \beta n dt)\ I(t-1)$. The authors derived a bias-correction formula for estimation of the number of infectious nodes at time t, $I(t)$, from the number of observed infectious nodes $I_d(t)$: Let σ be the average number of scans sent by an infectious node. After time interval dt, the expected number of

scans observed by k monitors is $\beta k I(t)\,\sigma dt/2^{32}$ (assuming the Internet is a complete graph), while the probability that any of the $I(t)-I_d(t)$ infectious nodes are observed is $1-(1-k/2^{32})^{\sigma\,dt}$. The worm is detected when the estimate of β starts oscillating around a positive constant value. Yet, it is not evident how the topology might affect Zou *et al.*'s detection mechanism. The study of Gu *et al.* [GSQ05] shows that the trend-detection mechanism faces challenges when a small number of monitored addresses (25,600 instead of 2^{20}) is used.

Chen *et al.* [CGK03] developed a deterministic approximation of model (6.3.1) for propagation on a complete graph K_n: If σ is the average scanning rate, with the assumption that the total number of nodes is 2^{32}, the average number of newly–infected nodes at step $(t+1)$ is $(S(t)-I(t))\left[1-(1-1/2^{32})^{\sigma\,I(t)}\right]$. If the rate of removal is γ, in the next time step $\gamma I(t)$ nodes will become infectious. Thus, the propagation can be described by a system of recurrences for the number of infectious and susceptible nodes. This model can be used to determine the size of the monitored space necessary for early detection of worms.

The assumption is that the worm typically scans some unassigned IP addresses or unused ports on assigned IP addresses. If there are k (inactive) monitored nodes, then the probability that one of them will be hit by a scan by time t is $P(t) = 1-(1-k/2^{32})^{\sigma\,I(t)\,-\,1}$. The authors conclude that when more than 2^{18} nodes are monitored, the system effectively detects and stops the propagation. Similarly, Wu

et al. [WVG04] develop detection architecture based on sensors monitoring the traffic at entry points for a sub–network and traffic going to unused addresses. The threshold detection algorithm is based on the number of newly–infected nodes, called *victim number*: An alarm is raised when continuous anomalies (*i.e.*, increases in the number of newly–infected hosts) are observed over a period of t_r time steps. The parameter t_r determines the sensitivity of the system to false alarms (the larger t_r, the greater probability that an actual attack will be detected, but also the time to react decreases; the smaller t_r, the greater the probability that a false alarm will be issued). This framework is validated by a simulation of Code Red worm attack, where the attack was detected when only 4% of susceptible nodes were infected, with 2^{16} monitored nodes.

The study of Gu *et al.* [GSQ05] shows that the victim–number algorithm can detect worm, employing random propagation strategy, even when the monitored network is of small size—25,600 nodes. However, the victim–number algorithm is ineffective in detection of worms that employ localized propagation strategies. Gu *et al.* [GSQ05] use the discrete approximation to model another detection mechanism: sliding window is kept for previous network traffic, and two general items are tracked—(1) for each port witnessed in the traffic, the address of the destination and scanning source (both from the monitored network) are recorded, (2) a counter, incremented each time a scan originates from a source that has previously received a scan on the same port.

Thus, at each time, the address of a possible victim and the number of scans sent from that address are available. The algorithm is implemented using three Bloom filters (two for destination addresses at times $(t-1)$ and t, and one for source addresses at time t). If the number of scans deviates from what is established as normal (during the training of the system), an alarm is raised and the host on that address is treated as a victim. This detection algorithm, unlike the victim–number algorithm, relies on active hosts (not only unassigned IP addresses) and can be used for detection of worms using localized propagation strategies.

6.3.2.2 Susceptible-Infectious-Removed Models of Dynamic Control Strategies

In this class of models, an infectious node can no longer spread the worm as a result of traffic-blocking or immunization. Here, at any time t, a node can be susceptible, infectious, or removed. Let γ be the rate at which infectious nodes are removed. Using analogous arguments as in the Susceptible-Infectious model, the general SIR model can be written as:

$$\frac{di(t)}{dt} = \beta\overline{d}(1 - i(t))\, i(t) - \gamma i(t),$$

$$\frac{dr(t)}{dt} = \gamma i(t), \tag{6.3.11}$$

with boundary conditions $i(0) = \frac{I(0)}{n} > 0$, $r(0) = \frac{R(0)}{n} \geq 0$, $s(t) + i(t) + r(t) = 1$. From the first equation in model (6.3.11), above, $\frac{di(t)}{dt} < 0$ if and only if $s(t) > \frac{\gamma}{\beta\overline{d}}$. Thus, if $s(t) > \frac{\gamma}{\beta\overline{d}}$, the fraction of infectious nodes decays exponentially.

Zou *et al.* [ZGT02] used a modification of the system (6.3.11) to determine the effect of a decreasing rate $\beta(t)$ and removal of susceptible and infectious nodes. This so-called *two-factor* model assumes complete graph as underlying topology, and a constant fraction of the removed-infectious nodes at any time t:

$$\frac{di(t)}{dt} = \beta(t)(1 - r(t) - r_s(t) - i(t))\, i(t) - \frac{dr(t)}{dt},$$

$$\frac{dr(t)}{dt} = \gamma i(t),$$

$$\frac{dr_s(t)}{dt} = \mu(1 - r(t) - r_s(t) - i(t))(i(t) + r(t)), \tag{6.3.12}$$

$$\beta(t) = \beta(0)(1 - i(t))^{\eta}.$$

It is unclear, however, how the parameters have been chosen in order to fit the data from the Code Red I worm propagation.

In another study, Zou *et al.* [ZGT03b], used model (6.3.11) to study *soft-quarantining* (here, quarantining means traffic-blocking). Every node (susceptible or infectious) can be quarantined individually when the worm detection mechanism raises alarm. The quarantine on a node is released after a quarantine time t_q, even if the node may be infectious. Let μ_i be the quarantine rate of an infectious node, and μ_s be the quarantine rate of a susceptible node. The probability that an infectious node is quarantined is $p_i = \dfrac{\mu_i t_q}{1 + \mu_i t_q}$, while the probability that a susceptible node is quarantined is $p_s = \dfrac{\mu_s t_q}{1 + \mu_s t_q}$. By assuming that changes of $R(t)$ and $I(t)$ are small during time t_q, one can find that the rate of infecting nodes is $\beta(1 - p_i)(1 - p_s)$. Zou *et al.* extended this model to include the case when only quarantined infectious nodes can be removed. The simulation study of the effect of the large quarantine time t_q concluded that the worm propagates faster compared to the prediction of the model. It should be noted that such a mechanism would be difficult to implement since it depends on every node being able to quarantine itself.

Boguna *et al.* [BPV03] studied dynamic immunization on scale-free topologies, via model (6.3.11) with $\gamma = 1$. Since a scale-free degree distribution is not concentrated around its mean value, the model must include differential equation for every group

of nodes of degree k. Let $\Theta\left(\{i_k(t)\}_{k=\delta}^{\Delta}\right)$ describes the probability that a susceptible node of degree k is adjacent to an infectious node. For a scale-free network, the probability that an edge is incident on a node of degree k is $kP(k)/\overline{d}$. The average probability that an edge is incident on an infectious node is then, $\Theta(t) = \frac{1}{\overline{d}} \sum_{k=\delta}^{\Delta} kP(k)\, i_k(t)$. The model can then be formulated as follows:

$$\begin{aligned} \frac{di_k(t)}{dt} &= \beta k\left(1 - i_k(t)\right)\Theta\left(\{i_k(t)\}_{k=\delta}^{\Delta}\right), \\ \frac{dr_k(t)}{dt} &= -i_k(t). \end{aligned} \tag{6.3.13}$$

which can be solved if one assumes that $i(0)$ is very small in the beginning of the propagation. In general, it is difficult to use model (6.3.13) in analyzing dynamic immunization since it does not provide an explicit set of differential equations that could be solved numerically.

6.4 Novel Near-optimal Dynamic Control Strategies

If a worm signature is not available, the control of propagation can be achieved in two ways: first, by reducing transmission from an infectious to each susceptible node (*i.e.*, quarantining), and, second, by limiting the number of susceptible nodes (*i.e.*,

by immunizing nodes through patch distribution). Although there is some work on isolating infectious nodes [LN04], [MVS03] through static content-filtering and address-blacklisting, much of the literature focuses on the second option and considers the design of immunization strategies. The problem of identifying the optimal control strategy, that combined immunization with quarantining, has not yet received attention.

In this section, we study five novel control strategies: (1) combination of static and dynamic immunization, (2) reactive dynamic immunization, (3) invariable dynamic immunization, (4) optimal soft-quarantining, and (5) predictive dynamic traffic-blocking. We use variations of the SIR model, described by equation (11), to study the effects of the first three control strategies, whereas, in the analysis of the predictive dynamic immunization, we employ individual–based simulation. The optimization parameter is the loss to the population, expressed through the number of removed susceptible nodes.

6.4.1 Combination of Static and Dynamic Immunization

The control strategy that combines static and dynamic immunization operates by immunizing (*i.e.*, removing) a portion of the susceptible nodes, prior to the beginning of the propagation. Let this portion be denoted by p (note that, $r(0) = p$). The

system of differential equations describing the dynamics of propagation (via $I(t)$) and the dynamics of control (via $R(t)$) is:

$$\frac{ds(t)}{dt} = -\beta\overline{d}s(t)\,i(t)\,,$$

$$\frac{di(t)}{dt} = \beta\overline{d}s(t)\,i(t) - \gamma i(t)\,, \tag{6.4.1}$$

$$\frac{dr(t)}{dt} = \gamma i(t)\,,$$

with boundary conditions $s(0) = \dfrac{S(0)}{n} - p \geq 0$, $i(0) = \dfrac{I(0)}{n} > 0$, $r(0) = p > 0$, $s(0) + i(0) + r(0) = 1$.

Realistic deployment of this control strategy requires patching of susceptible nodes (once vulnerability is detected, but before the propagation starts) and a self-replicating agent that could block the traffic from infectious nodes (detected via the worm signature).

6.4.2 Reactive Dynamic Immunization

In this variation of dynamic immunization, a portion of susceptible nodes proportional to the number of infectious nodes is removed, at rate μ, during the propagation. Realistic deployment of this strategy would require a self-replicating agent that patches susceptible nodes and blocks the traffic of infectious nodes. The model is then given by:

$$\begin{aligned} \frac{ds\,(t)}{dt} &= -\beta\bar{d}s\,(t)\,i\,(t) - \mu i\,(t)\,, \\ \frac{di\,(t)}{dt} &= \beta\bar{d}s\,(t)\,i\,(t) - \gamma i\,(t)\,, \\ \frac{dr\,(t)}{dt} &= (\mu+\gamma)\,i\,(t)\,. \end{aligned} \tag{6.4.2}$$

Reactive dynamic immunization will have the same effects as the combination of static and dynamic immunization when the fraction p of susceptible nodes removed before the propagation starts equals $1 - \dfrac{\gamma\left(\beta\bar{d}+\mu\right)}{\beta\bar{d}\,(\gamma+\mu)}$. It is interesting to study the relationship between the rates μ and γ, and its effect on the total number of infectious and removed nodes at the end of the propagation, as shown in Section 6.5.

6.4.3 Invariable Dynamic Immunization

The limited ability of the quarantining system to distribute patches can be modeled by a constant rate of immunizing susceptible nodes. The system of differential equations is similar to model (6.4.2), and can be formally written as:

$$\begin{aligned} \frac{ds\,(t)}{dt} &= -\beta\overline{d}s\,(t)\,i\,(t) - \mu, \\ \frac{di\,(t)}{dt} &= \beta\overline{d}s\,(t)\,i\,(t) - \gamma i\,(t)\,, \\ \frac{dr\,(t)}{dt} &= \gamma i\,(t) + \mu. \end{aligned} \tag{6.4.3}$$

We note that for system (6.4.3) there is no closed–form solution for the total number of removed nodes at the end of the propagation.

6.4.4 Optimal Soft-quarantining

The effectiveness of a control policy, modeled by the SIR, is usually measured in terms of its ability to reduce the average number of new infections produced by an infectious node (during its infectious period) if placed in a population of susceptible nodes. This quantity, known as the basic reproductive ratio R_0, can be expressed as the ratio β/γ,

when the network is a complete graph. It is well known [AM92] from epidemiological studies that when $R_0 > 1$, the number of infections will grow, whereas if $R_0 < 1$, the new infections, on average, will decline and major epidemics cannot occur. Therefore, any control policy aims at reducing the value of R_0 below one. There are, however, further potential requirements for a control policy—for instance, spatial containment of the propagation, reduction of the propagation duration, minimization of overall losses to the population, or a combination of these requirements. Interesting, and, yet not investigated are strategies that minimize losses to the population.

Our soft quarantining control strategy is a combination of two parts: (1) dynamic immunization of infectious nodes when detected, and (2) dynamic quarantining of nodes whose history (*e.g.*, recent established connections) suggests an enhanced risk for getting infected. Both immunized and quarantined nodes will be considered removed. This strategy prevents further transmission from infectious nodes, but may also result in removal of some susceptible nodes. This leads to a trade-off: increased levels of control result in a greater reduction in transmission, but also in an increase in the number of removed susceptible nodes. As mentioned in Section 6.3.2.2, *soft quarantining* with duration t_q has already been studied by Zou [ZGT03b]. Here, we focus on the optimal level of a control strategy that combined quarantining and immunization, when $t_q \to \infty$, in order to minimize the overall losses to the population expressed through the number of removed nodes.

We consider the SIR model, where removed nodes arise from: (1) quarantining of detected infectious nodes at rate γ, and (2) quarantining, at rate c, of nodes that has not yet been identified as infectious but whose history shows that they are at greater risk of getting infected. Implementation of quarantining policy (2) will remove a fraction f of nodes at risk from a given infectious nodes; since increases levels of control (*i.e.*, larger c) may require greater fraction of susceptible nodes to be removed, we assume that f is a function of c. In turn, this will result in probability $\beta\,(\,1\,-\,f\,(c)\,)\;s\,(t)$ of finding a susceptible node that has not been quarantined. The model on a graph, whose average degree is $\overline{d}$, can be written as:

$$\begin{aligned}
\frac{ds\,(t)}{dt} &= -\beta\overline{d}\,(1-f\,(c))\,s\,(t)\,i\,(t) - c\overline{d}s\,(t)\,i\,(t)\,, \\
\frac{di\,(t)}{dt} &= \beta\overline{d}\,(1-f\,(c))\,s\,(t)\,i\,(t) - \gamma i\,(t)\,, \\
\frac{dr\,(t)}{dt} &= c\overline{d}s\,(t)\,i\,(t) + \gamma i\,(t)\,.
\end{aligned} \tag{6.4.4}$$

The general results about the SIR model 6.4.4 are [AM92]:

1. An epidemic can occur only when $R_0 = \dfrac{\beta\,(1-f\,(c))\;\overline{d}}{\gamma} \quad > 1,$
2. Function $S\,(t)$ is monotonically decreasing, $R\,(t)$ is monotonically decreasing, and $I\,(t)$ is unimodal (has one maximum),

3. The epidemic eventually dies out, with some proportion of susceptible remaining, given by:

$$s(t) = e^{(s(t)-1)R_0}, \quad t \to \infty.$$

Let the number of nodes quarantined during an infectious period of an infectious node be denoted by $p = c/\gamma$. The final fraction of the removed nodes can be determined by dividing the first by the second equation from the system (6.4.4) and integrating over I, to obtain:

$$R_0 \left(S(\infty) - S(0)\right) - n \ln \frac{S(\infty)}{S(0)} = \left(-R_0 - p\right)\left(I(\infty) - I(0)\right). \tag{6.4.5}$$

Further, by employing the final relationship from model (6.4.4), one can get:

$$\ln \frac{n - R(\infty) - I(\infty)}{n - I(0)} = R_0 \frac{I(0) - R(\infty) - I(\infty)}{n} + (R_0 + p) \frac{I(\infty) - I(0)}{n},$$

or equivalently:

$$r(t) = 1 - (1 - i(0))\, e^{-R_0\, r(t) - p\, i(0)}, \; t \to \infty. \tag{6.4.6}$$

Since the fraction of removed nodes that are expected to turn into infectious (and could spread the worm in case they were not quarantined) is $\dfrac{R_0}{R_0 + p}$, the final fraction

of infectious quarantined nodes can be expressed as:

$$r(t) - r_s(t) = \frac{R_0}{R_0 + p}\left(r(t) - i(0)\right) + i(0)\,,\ t \to \infty. \tag{6.4.7}$$

Theorem 6.4.1. *The total fraction of removed nodes, at the end of propagation, decreases with the increase of the control parameter c, if $\beta \dfrac{df(c)}{dc} > \dfrac{i(0)}{r(t)}$.*

Proof. The function $r(t)$, $t \to \infty$ depends on c, through $f(c)$ and $p = c/\gamma$. Therefore, we investigate how $r(t)$, $t \to \infty$, changes in respect to the increase of c. To locate the minimum, we look at the conditions under which $\dfrac{dr(t)}{dc} < 0$. By differentiating (6.4.6) with respect to c, one obtains:

$$\frac{dr(t)}{dc} = \frac{r(t)}{\gamma}\ \left(1 - i(0)\right) e^{-R_0\, r(t) - pi(0)} \left[\frac{i(0)}{r(t)} - \beta \frac{df(c)}{dc}\right]. \tag{6.4.8}$$

The sign of $\dfrac{dr(t)}{dc}$ is determined by the last multiplicand of the right–hand side in equation (6.4.8). It follows that $\dfrac{dr(t)}{dc} < 0$, if and only if $\beta \dfrac{df(c)}{dc} > \dfrac{i(0)}{r(t)}$. □

To conclude: (1) the analysis of model (6.4.4) shows that the amount of losses in the population are determined by the function $f(c)$, expressing the fraction of removed nodes that are at risk from an infectious node. This is directly quantified

through the change of the basic reproductive ratio $R_0 = \frac{\beta (1 - f(c)) \overline{d}}{\gamma}$, and (2) the first derivative $\frac{df(c)}{dc}$, of the function $f(c)$, determines whether or not an increase in the value of c will cause a decrease in the final fraction of removed nodes $r(t)$—as stated by Theorem 6.4.1.

6.4.5 Predictive Dynamic Traffic-blocking

In realistic deployment, the previous three control strategies require a patch, for immunization of susceptible nodes, and the worm signature, for elimination of the worm on the infectious nodes. The predictive dynamic traffic-blocking could be applied when *neither patch nor worm signature is available.* This control strategy employs information about the size and the behavior (anomalous or normal) of the nodes in a local neighborhood. The predictive dynamic traffic-blocking could be thought of as a realization of the optimal control strategy analyzed in Section 6.4.4. Based on the available information, the predictive strategy assesses the *risk* for a node to become infectious. The *risk* for a node to become infectious is described as a function of probability of becoming infectious and the consequence of being infectious. The probability for a node to become infectious is determined by the number of infectious nodes in the local neighborhood of the node. The consequence of being infectious

is determined by the degree of the node—the higher the degree, the greater the effect of propagation from that node. Before presenting the detailed description of the predictive dynamic traffic-blocking, we give some definitions:

Definition 6.4.1. For a node u and an integer l, the local neighborhood of u, denoted by $N(u)$, is composed of all nodes whose distance from u is no greater than $l, i.e.$

$$N(u) = \{v : d(u, v) \leq l\}.$$

Note that $P(u) = N(u) = \bigcup_{j=1}^{l} N_j(u)$, where the jth–neighborhood $N_j(u) = \{v : d(u, v) = j\}$. The set of infectious nodes in $N_j(u)$ will be denoted by $N_j^i(u)$.

Definition 6.4.2. The susceptibility of a node u, denoted by $\eta(u)$, is the weighted sum of the cardinalities of all neighborhoods $N_j(u)$, $1 \leq j \leq l$, *i.e.*,

$$\eta(u) = \sum_{j=1}^{l} \frac{1}{j} |N_j(u)|.$$

Definition 6.4.3. Given a node u, let $\bigcup_{j=1}^{l} N_j^i(u)$ be the set of infectious nodes in $N(u)$. The risk of u becoming infectious, denoted by $\rho(u)$, is the weighted sum

$$\rho(u) = \frac{d(u)}{n} \frac{\sum_{j=1}^{l} \frac{1}{j} \left|N_j^i(u)\right|}{\sum_{j=1}^{l} \frac{1}{j} |N_j(u)|}.$$

Addition of one infectious node in $N(u)$ say at distance $k, 1 \leq k \leq l$, increases the risk by

$$\frac{d(u)}{n} \frac{1}{k \sum_{j=1}^{l} \frac{1}{j} |N_j(u)|}. \tag{6.4.9}$$

Remark 6.4.1. Note that $\frac{\sum_{j=1}^{l} \frac{1}{j} \left|N_j^i(u)\right|}{\sum_{j=1}^{l} \frac{1}{j} |N_j(u)|}$ represents the probability for the node u to become infectious, while $\frac{d(u)}{n}$, the ratio of the degree of u to the total number of nodes, expresses the consequence of node u being infectious.

The predictive dynamic traffic blocking is described as follows: Let τ be a given threshold, c be the number of susceptible nodes on which the strategy is applied, and ϑ be a counter of new infectious nodes. Predictive dynamic traffic-blocking is a greedy algorithm that selects c susceptible nodes with *highest risk* of being infected, once the counter ϑ exceeds the value of τ. Whenever an infectious node v is detected, the risk of every susceptible node u in $P(v)$ is updated, by using equation (6.4.9). The algorithm is formally given in Figure 6.1, below.

One can think of different factors, representing the consequence of infection, that could be included in the function $\rho(u)$: the type of the node—host, gateway, server, or router, the amount of traffic that passes through it, or a combination of these. In Section 6.5, we point out that even without traffic data, the function $\rho(u)$ specified

Algorithm *Predictive Dynamic Control*

Input:

G, graph
l, integer
c, number of susceptible nodes to be removed
V_I, set of infected node

Output:

V_{R_s}, list of susceptible nodes

```
for every node u ∈ V(G) − V_I do
    calculate ρ(u)
    L ← L ∪ {(u, ρ(u))}
end for
sort L in decreasing order of ρ(u)
return the first c elements of L
```

Figure 6.1: Greedy algorithm for predictive dynamic control

in terms of graph–theoretic characteristic (*i.e.*, the degree of node u) performs well. For the empirical analysis, we will use local neighborhood with $l = 2$ that models limited information about the network environment.

6.5 Analysis of the Proposed Control Strategies

In this section, we present the comparative analysis of four near-optimal control strategies described in Section **??** on two types of graphs of same order: (1) Internet graphs and (2) Barabasi–Albert scale-free graphs.

We combined the individual-based simulation of the stochastic propagation process with simulation of the control strategy. Such simulation provides very precise and detailed information about the dynamics of the control strategy. The scheduler (of the simulation) is implemented as a priority queue (just like in Section 5.4). The system is composed of nodes that can be either susceptible, infectious, or removed. There are two types of events that can take place: infection and removal. If a node u is infectious, it attempts infection of each of its neighbors at rate β. Node u is removed, in the SIR model, at rate at rate γ. Let node u be removed at time t. Any infection generated by node u in the time period after t is discarded by the scheduler. The event of node u attempting infection of an infectious or a removed neighbor neighbor v at time t is also discarded by the scheduler. The removal process (carried at rate μ, on susceptible nodes) could be thought of as another birth process that affects the state of the nodes. Thus, the same rules apply as in the previously described cases.

Simulations were carried on the Macroscopic Internet graphs shown in Figure 5.1 and on topologies generated by the Barabasi-Albert model of same order as the graphs in Figure 5.1. With the help of the individual-based simulation, we are able to answer the following question regarding each control strategy described in Section 6.4:

1. The maximum number of infectious nodes (averaged over 5000 simulations),

2. The necessary number of immunizations (node-removals) to contain the propagation,

3. The time t_e required for containment (averaged over 5000 simulations).

4. The number of susceptible nodes at time t_e

It is interesting to point out that the analysis of the third problem does not only involve the average time required for containment, but also the tail of the distribution of infectious nodes over time. Distributions, in which, at a large time moment, the number of infectious nodes is small, (*i.e.*, distributions with longer tails) can have considerable impact on worm propagation and control. This effect is particularly strong for distributions whose tail include nodes of higher degrees, as concluded in Chapter 5. Here, for a given control strategy, the fraction of susceptible nodes at the end of propagation (at time moment t_e) is used as a measure of its effectiveness.

Two sets of experiments were performed for each graph and each control strategy: in the first set, the propagation was initiated at a node of minimum degree, while in the second set, the propagation was initiated at a node of maximum degree. The latter is of particular importance, because it can be use to estimate the effectiveness of a control strategy when the worm has the biggest probability to a wide-spread propagation in shortest time. Moreover, in each set of experiment, for given set of simulation parameters (β, γ, and μ), we performed a simple statistical analysis of the results—number of infectious nodes, number of removed nodes, and minimum time required for containment—in order to estimate the effect of stochastic fluctuations.

The comparison of the proposed control strategies for the graph from 06.13.2001 with 10515 nodes is shown in Figures 6.6, 6.7, 6.8, and 6.9.

Clearly, there exists a *stochastic ordering* of the control strategies (ordering in terms of the $r.v.$ representing the number of infectious nodes over time), although we do not make this precise. The dynamic control, modeled by the classical Susceptible-Infectious-Removed framework, performs worse than the invariant dynamic immunization, which in turn has worse performance compared to the control strategy that combines static and dynamic immunization. However, the tail of the distribution of infectious nodes for the strategy that combines static and dynamic immunization is longer compared to that of the invariable dynamic immunization. The predictive dynamic control strategy outperforms the rest of the proposed control techniques, in terms of both, the number of infectious nodes and the number of removed nodes at the end of the propagation (at time moment t_e).

Detailed statistical analysis of results concerning the invariable dynamic immunization and the predictive traffic-blocking on four Macroscopic Internet graphs is shown in Figures 6.2, 6.3, 6.4, 6.5. Regression analysis of the obtained results shows that doubling the removal rate μ results in an increase of the number of susceptible nodes at the end of the propagation by 4 times. Moreover, only substantial increase (of order 100) of the rate μ, as compared to β, results in increasing the total number of susceptible nodes at time t_e.

Graph AS 02.07.2000								
β	μ	average $I(t_e)$	average $R(t_e)$	average t_e	$\sigma I(t_e)$	$\sigma R(t_e)$	$\sigma\ t_e$	% $S(t_e)$
1.80	3.60	5272.340	2674.080	3.276	1106.107	1079.993	0.260	0.120%
1.80	0.90	6432.480	1521.980	5.005	1258.377	1222.918	1.662	0.019%
1.80	0.30	7187.440	768.300	6.991	701.966	697.929	5.094	0.003%
1.50	3.60	5133.760	2811.640	4.057	935.860	772.113	0.633	0.133%
1.50	0.90	6283.800	1669.540	5.114	1054.694	976.662	0.965	0.033%
1.50	0.30	7074.600	881.060	8.393	951.347	936.792	10.723	0.004%
0.90	3.60	4774.140	3165.340	6.602	746.164	627.453	1.883	0.208%
0.90	0.90	5845.120	2106.060	7.237	731.210	691.731	1.409	0.061%
0.90	0.30	6736.060	1218.780	11.194	1190.792	1173.032	10.434	0.015%
0.50	3.60	4457.520	3481.620	11.687	517.438	451.547	4.297	0.212%
0.50	0.90	5352.540	2594.800	12.384	913.968	811.469	5.109	0.109%
0.50	0.30	6288.340	1665.640	15.354	716.596	688.847	12.685	0.025%

Graph AS 02.07.2000		$c = 3$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	1273.520	3817.560	3.805	94.355	283.064	0.728	36.010%
1.5	1252.580	3754.740	4.315	75.850	227.550	0.853	37.062%
0.9	1255.720	3764.160	7.847	85.374	256.122	1.603	36.904%
0.5	1275.440	3823.320	13.574	109.696	329.088	2.101	35.913%
0.2	1251.740	3752.220	33.414	77.219	231.658	5.421	37.105%

Graph AS 02.07.2000		$c = 4$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	343.320	1369.280	2.778	90.432	361.727	0.607	78.474%
1.5	358.820	1431.280	3.324	92.031	368.126	0.764	77.500%
0.9	346.640	1382.560	5.624	85.556	342.225	1.321	78.265%
0.5	363.620	1450.480	9.924	71.175	284.700	2.435	77.198%
0.2	343.460	1369.840	25.779	87.913	351.650	6.972	78.465%

Figure 6.2: Statistical analysis of the invariable dynamic immunization and the predictive dynamic traffic-blocking strategy on the Macroscopic Internet graph from 02.07.2000, where propagation starts at node of degree 1772

In light of Theorem 6.4.1, we also study who the increase in the number, c, of susceptible nodes to which traffic-blocking is applied, might improve the effectiveness of the predictive dynamic traffic-blocking. For the case when $\tau = 1$, *i.e.*, the predictive dynamic traffic-blocking is applied each time an infectious node is detected, $c = 4$ is the optimum value. For this value of c, the predictive dynamic traffic-blocking results

Graph AS 02.10.1999								
β	μ	average $I(t_e)$	average $R(te)$	average te	$\sigma I(te)$	$\sigma R(te)$	$\sigma\ te$	% $S(te)$
1.80	3.60	3514.300	1834.400	3.314	837.480	777.388	0.490	0.155%
1.80	0.90	4306.340	1049.140	4.517	610.311	593.429	1.152	0.028%
1.80	0.30	4812.680	544.020	6.980	498.344	500.755	8.926	0.006%
1.50	3.60	3414.500	1931.240	4.043	516.173	479.370	0.368	0.210%
1.50	0.90	4200.180	1154.420	5.177	595.498	549.228	1.656	0.045%
1.50	0.30	4744.620	611.920	7.772	323.016	324.157	6.925	0.009%
0.90	3.60	3180.540	2163.840	6.618	540.335	398.586	2.548	0.236%
0.90	0.90	3902.300	1450.780	7.434	801.194	744.910	2.308	0.073%
0.90	0.30	4515.640	840.300	10.681	519.419	502.745	13.176	0.020%
0.50	3.60	2964.440	2374.980	11.136	307.680	325.489	3.287	0.328%
0.50	0.90	3569.620	1779.220	11.288	655.832	553.808	4.934	0.152%
0.50	0.30	4195.040	1159.480	15.505	1100.733	1039.316	18.528	0.046%

Graph AS 02.07.1999		$c = 3$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	792.060	2373.180	3.419	71.619	214.857	0.669	40.914%
1.5	778.240	2331.720	4.076	78.158	234.473	0.942	41.946%
0.9	774.000	2319.000	6.812	67.461	202.384	1.260	42.262%
0.5	782.640	2344.920	12.090	82.756	248.268	2.028	41.617%
0.2	791.980	2372.940	31.161	71.657	214.971	5.524	40.920%

Graph AS 02.07.1999		$c = 4$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	637.000	2544.000	3.391	41.776	167.105	0.701	40.620%
1.5	635.320	2537.280	4.119	34.915	139.659	0.766	40.777%
0.9	643.640	2570.560	6.936	40.298	161.191	1.378	40.000%
0.5	632.940	2527.760	11.795	37.840	151.362	1.717	40.999%
0.2	643.480	2569.920	31.037	35.366	141.466	6.459	40.015%

Figure 6.3: Statistical analysis of the invariable dynamic immunization and the predictive dynamic traffic-blocking strategy on the Macroscopic Internet graph from 02.07.1999, where propagation starts at node of degree 1193

in the minimum number of removed nodes and a fraction of susceptible nodes greater than 40% at the end of the propagation.

Graph AS 02.10.1998								
β	μ	average $I(t_e)$	average $R(t_e)$	average t_e	$\sigma I(t_e)$	$\sigma R(t_e)$	σt_e	% $S(t_e)$
1.80	3.60	2700.820	1466.880	3.314	630.926	428.842	0.331	0.294%
1.80	0.90	3308.660	868.340	4.631	630.107	569.331	1.179	0.072%
1.80	0.30	3735.940	443.660	7.226	456.833	453.698	8.996	0.010%
1.50	3.60	2621.160	1544.600	3.833	417.688	348.041	0.377	0.341%
1.50	0.90	3235.160	941.300	5.305	549.770	496.296	1.648	0.085%
1.50	0.30	3671.600	507.700	8.415	364.490	357.439	10.481	0.017%
0.90	3.60	2445.500	1716.020	6.624	321.847	237.530	1.438	0.442%
0.90	0.90	3002.660	1170.840	7.322	532.515	415.688	1.671	0.156%
0.90	0.30	3483.800	694.880	11.052	701.102	695.944	7.427	0.032%
0.50	3.60	2292.920	1869.180	11.253	320.361	217.171	4.812	0.428%
0.50	0.90	2735.060	1432.220	12.222	673.772	523.114	5.127	0.304%
0.50	0.30	3231.020	945.600	15.491	633.081	563.143	12.303	0.081%

Graph AS 02.10.1998		$c = 3$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	631.380	1891.140	3.477	37.998	113.995	0.761	39.653%
1.5	646.020	1935.060	4.165	45.621	136.863	0.764	38.252%
0.9	630.140	1887.420	6.981	43.265	129.796	1.312	39.771%
0.5	650.300	1947.900	13.361	55.005	165.015	2.047	37.842%
0.2	640.100	1917.300	30.576	58.011	174.034	7.154	38.818%

Graph AS 02.10.1998		$c = 4$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	523.820	2091.280	2.924	129.724	518.895	0.652	37.438%
1.5	554.540	2214.160	3.607	129.287	517.150	0.884	33.763%
0.9	527.840	2107.360	5.757	155.817	623.269	1.202	36.957%
0.5	478.200	1908.800	11.697	181.106	724.425	2.820	42.895%
0.2	572.960	2287.840	27.823	121.856	487.425	5.212	31.560%

Figure 6.4: Statistical analysis of the invariable dynamic immunization and the predictive dynamic traffic-blocking strategy on the Macroscopic Internet graph from 02.10.1998, where propagation starts at node of degree 879

6.6 Summary

Despite the recent surge of research in control of worm propagation, currently, there is no effective defense system against such cyber attacks. Here, we first present a classification of existing control strategies in two groups—static and dynamic—and discussed their advantages and disadvantages. As propagation and control strategies are

Graph AS 08.11.1997								
β	μ	average $I(te)$	average $R(te)$	average te	$\sigma I(te)$	$\sigma R(te)$	$\sigma\ te$	% $S(te)$
1.80	3.60	1912.220	1090.340	3.529	507.196	333.453	0.421	0.413%
1.80	0.90	2355.300	656.400	4.737	504.990	416.286	1.442	0.109%
1.80	0.30	2675.080	339.620	7.542	312.075	306.934	15.583	0.010%
1.50	3.60	1855.660	1142.260	3.876	493.413	198.115	0.405	0.567%
1.50	0.90	2299.760	711.820	5.425	668.839	585.212	1.489	0.113%
1.50	0.30	2625.220	388.800	8.654	720.175	647.633	11.389	0.033%
0.90	3.60	1742.620	1257.320	6.285	495.383	288.875	1.681	0.500%
0.90	0.90	2129.640	878.920	7.611	656.031	466.157	1.383	0.214%
0.90	0.30	2484.700	528.220	11.631	1046.255	918.093	19.370	0.069%
0.50	3.60	1630.800	1364.460	10.907	245.388	163.764	3.759	0.655%
0.50	0.90	1944.460	1057.380	12.083	562.580	322.036	4.325	0.436%
0.50	0.30	2302.180	708.820	15.122	752.069	584.477	10.184	0.133%

Graph AS 08.11.1997		$c = 3$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	381.480	1141.440	2.855	56.576	169.728	0.715	49.489%
1.5	390.720	1169.160	3.289	61.850	185.551	0.663	48.263%
0.9	390.860	1169.580	5.490	54.749	164.246	0.860	48.244%
0.5	378.820	1133.460	10.077	58.260	174.781	2.404	49.841%
0.2	383.540	1147.620	24.050	56.458	169.375	5.870	49.215%

Graph AS 08.11.1997		$c = 4$					
β	average $I(T)$	average $R(T)$	average T	$\sigma I(T)$	$\sigma R(T)$	σT	% $S(T)$
1.8	313.040	1248.160	2.801	47.803	191.213	0.619	48.219%
1.5	311.400	1241.600	3.416	44.148	176.591	0.824	48.491%
0.9	303.580	1210.320	5.798	36.576	146.304	1.516	49.788%
0.5	312.200	1244.800	10.061	46.318	185.273	3.096	48.358%
0.2	310.595	1238.378	24.769	46.609	186.437	5.374	48.624%

Figure 6.5: Statistical analysis of the invariable dynamic immunization and the predictive dynamic traffic-blocking strategy on the Macroscopic Internet graph from 08.11.1997, where propagation starts at node of degree 590

tightly coupled with a particular detection mechanism, the review of control strategies also encompasses the existing detection mechanism. Our thorough survey of worm control and detection mechanisms shows that there are no control strategies that use local network–information. Since the problem of containing worms is NP-hard, as concluded in Chapter 4, we presented five novel control strategies—(1) dynamic soft-quarantining, (2) combination of static and dynamic immunization reactive dynamic

immunization, (3) invariable dynamic immunization, (4) soft-quarantining, all modeled within the Susceptible-Infectious-Removed epidemiological framework, and (5) predictive dynamic traffic-blocking that employs limited information about the network topology and the level of worm propagation. For the dynamic soft-quarantining, we determined a condition that guarantees minimum number of removed nodes at the end of propagation. Like in Chapter 5, the analysis of the proposed control strategies is carried out with the help of the epidemiological approach, with information about the network structure, and individual-based simulation. Simulation results show that there exists a *stochastic ordering* of the control strategies: The dynamic control, modeled by the classical Susceptible-Infectious-Removed framework, performs worse than the invariant dynamic immunization, which in turn has worse performance compared to the control strategy that combines static and dynamic immunization. The predictive dynamic control strategy outperforms the rest of the proposed control techniques, in terms of both, the number of infectious nodes and the number of removed nodes at the end of the propagation (at time moment t_e). For c, the number of susceptible nodes removed per new infectious node, of value 4, the predictive dynamic traffic-blocking results in minimum number of removed nodes, in line with the analysis of dynamic soft-quarantining, and a fraction of susceptible nodes greater than 40% at the end of the propagation.

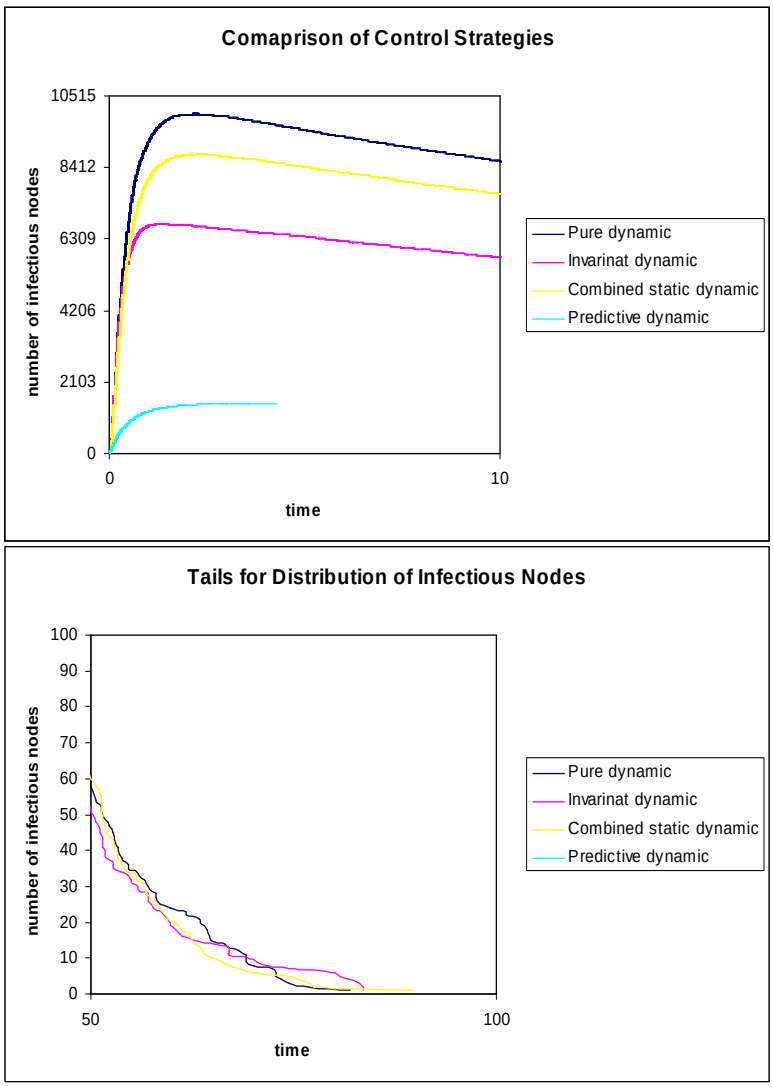


Figure 6.6: The number of infectious nodes and the tail of its distribution over time for four control strategies: (1) dynamic Susceptible–Infectious–Removed, (2) invariable dynamic immunization, (3) combination of static and dynamic immunization, and (4) predictive traffic-blocking with parameters $\beta = 1.8, \gamma = 0.02, p = 0.1, \mu = 3.6, c = 4$ on Macroscopic Internet graph from 06.13.2001

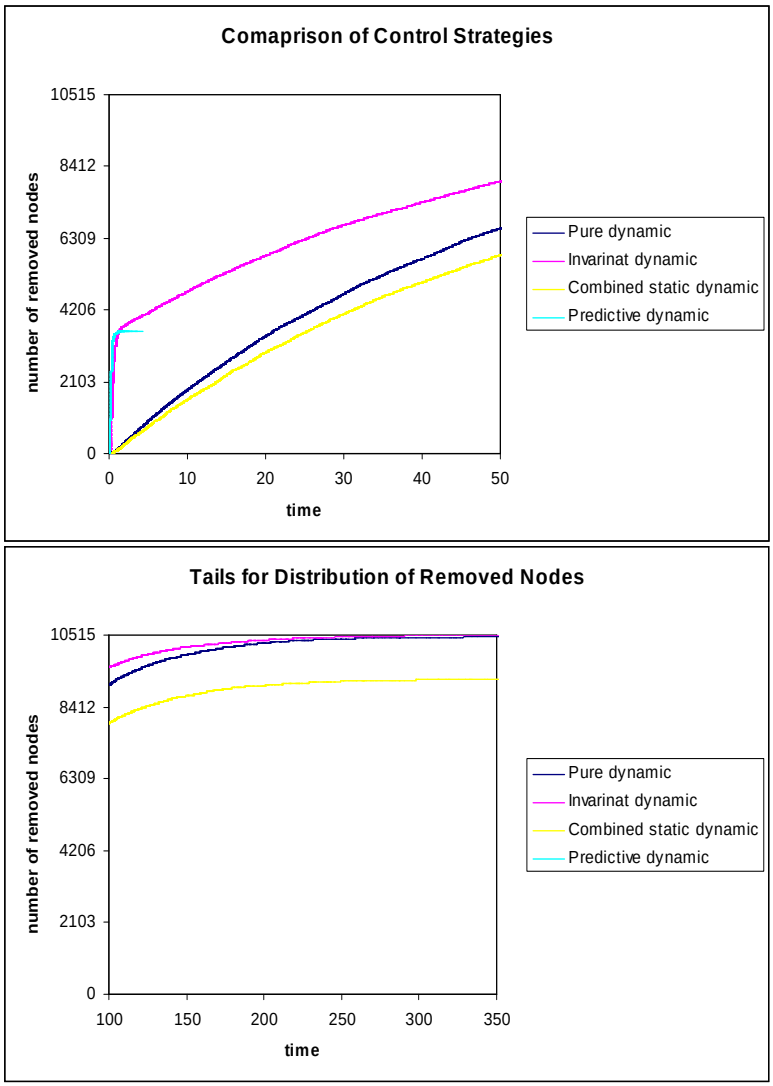


Figure 6.7: The number of infectious nodes and the tail of its distribution over time for four control strategies: (1) dynamic Susceptible–Infectious–Removed, (2) invariable dynamic immunization, (3) combination of static and dynamic immunization, and (4) predictive traffic-blocking with parameters $\beta = 1.8, \gamma = 0.02, p = 0.1, \mu = 3.6, c = 4$ on Macroscopic Internet graph from 06.13.2001

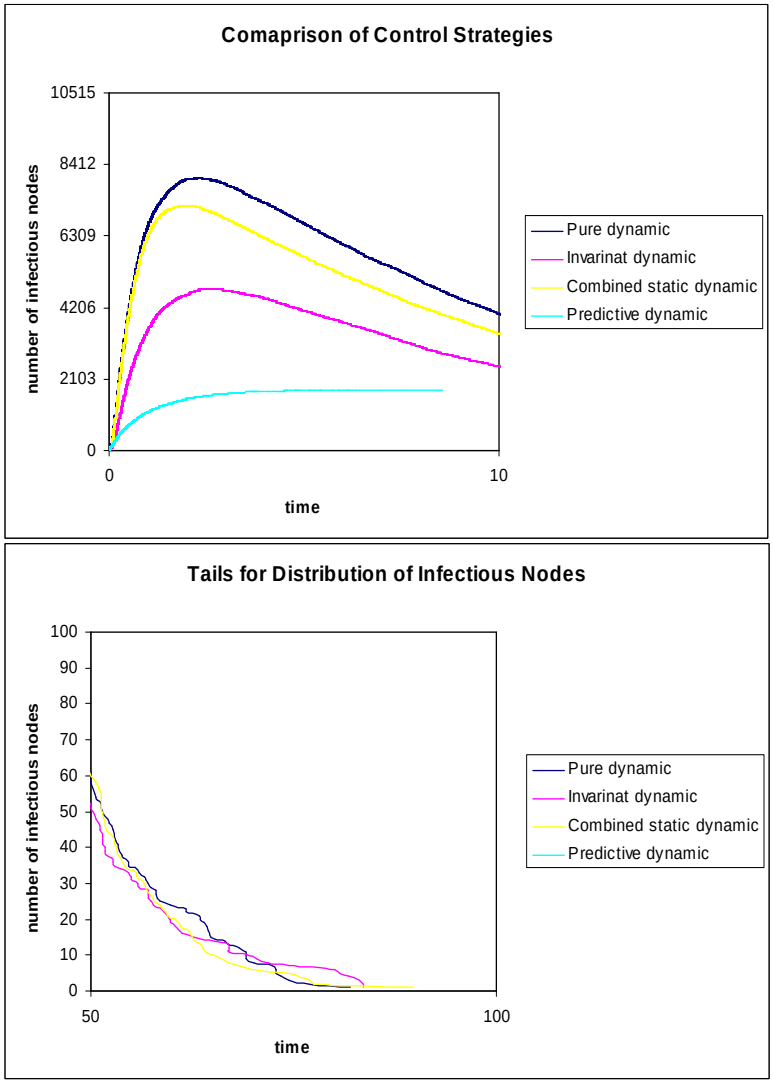


Figure 6.8: The number of infectious nodes and the tail of its distribution over time for four control strategies: (1) dynamic Susceptible–Infectious–Removed, (2) invariable dynamic immunization, (3) combination of static and dynamic immunization, and (4) predictive traffic-blocking with parameters $\beta = 0.9, \gamma = 0.1, p = 0.3, \mu = 0.9, c = 4$ on Macroscopic Internet graph from 06.13.2001

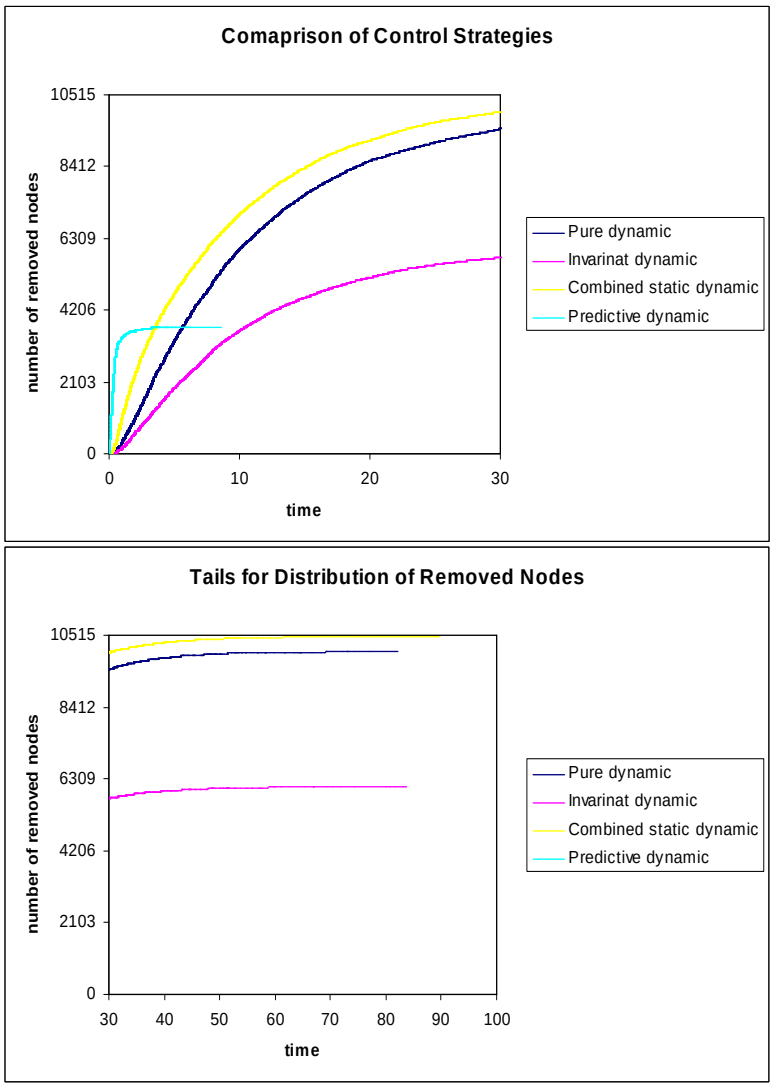


Figure 6.9: The number of infectious nodes and the tail of its distribution over time for four control strategies: (1) dynamic Susceptible–Infectious–Removed, (2) invariable dynamic immunization, (3) combination of static and dynamic immunization, and (4) predictive traffic-blocking with parameters $\beta = 0.9, \gamma = 0.1, p = 0.3, \mu = 0.9$ on Macroscopic Internet graph from 06.13.2001

CHAPTER 7

DETECTION VIA DISTRIBUTED BLACKHOLES

7.1 Introduction

A necessary requirement for any control strategy is timely and accurate detection of the worm. Automated detection of cyber attacks employing network worm is performed using an *Intrusion Detection System* (*IDS*) that attempts to detect the existence of an attack, isolate its source, and inform the system (network) administrator. There are two types of IDSs: signature–based and anomaly–based [PP03]. *Signature–based* detection performs pattern–matching of binary code or a series of commands (and events) that is known to indicate a particular intrusion. This type of IDS scans network packets or examines audit records generated by the operating system looking for such signatures. For instance, signature–based IDS may look for

network packets that are directed to ports or services that are known to be vulnerable. Instead of looking for matches, *anomaly-based* detection uses aberrations from the system's behavior that is considered normal (based on the knowledge of behavioral patterns) to indicate possible attacks.

The *sensitivity* of an IDS can be expressed by the number of false positives and false negatives it produces. *False positive* indicates the event of signaling an attack, when an attack has not happened. Large number of false positives increases the possibility of ignoring some signals that might be associated with a real attack. *False negative* indicates that real attacks pass through the IDS undetected. Therefore, developing an IDS with minimal number of false negatives and false positives is of particular importance.

The chapter is organized as follows: Section 2 gives overview of existing worm detection mechanisms with their advantages and disadvantages. In Section 3, the novel Detection via Distributed Blackholes is described in detail. Finally, Section 4 presents the combination of the distributed architecture and algorithm with the control strategy called contact-tracing.

7.2 Existing Detection Techniques

A prerequisite for a defense system is its ability to identify a worm signature—a string of bytes in the traffic that passes through a network link. Most techniques for detection of network worms require the use of attack–free data (that can be leveraged to train an anomaly detector), a large block of unused IP addresses for eliciting erratic network (or host) behavior, or a recently developed honeypot approach.

7.2.1 Loss of Self–similarity of Network Traffic

Paxon and Floyd's study [PF95] of the number of bytes and duration of a session for several different applications (telnet, e-mail, FTP) showed that none of the considered quantities was well-modeled as a Poisson process, but followed the log-extreme, log-normal, or Pareto distribution. Leland, Taqqu, Willinger, and Wilson [LTW94] were the first to demonstrate that the distribution of traffic in LANs and on the Internet exhibits self-similarity.

Schleifer and Mannle [SM01] proposed a method for detecting attacks based on the changes in the self-similarity of the network traffic. Cabrera, Ravichandran, and Mehra [CRM00] showed that denial-of-service and scanning/probing can affect network traffic to a measurable extent. Li, Jia, and Zhao [MZ01] demonstrated

mathematically that a significant change in the Hurst parameter (from that of the non-attack traffic) can be used to detect an attack. Allen and Marin [AM03] indicated that the loss of self-similarity can be used to detect possible denial-of-service attack or intense probing, provided only that the normal (non-attack) traffic is self-similar. They propose a procedure LOSS that divides the time-line (for the examined traffic) into overlapping one-hour windows using 15-minuite increments for which the Hurst parameter is calculated; this could be used in determining more accurately the onset and the ending of detected attacks.

7.2.2 Abnormal Behavior at the Source of Attack

With the increase of the network components' speed, it is becoming harder for the IDSs and firewalls to process packets fast enough to ensure full protection. While traditional IDS operate at the destination of the attack, *behavior blocking* prevents a computer from performing malicious activities by *blocking* (*e.g.*, slowing, turning off) its attacking capabilities. This technique, similarly to anomaly-based IDSs, uses a policy for the allowed behaviors of an application, and any infringements of that policy are detected and reported. Bruschi and Rosti [BR00] call this technique *host disarming*, implemented through filtering components added as middleware between the device drivers and the kernel (so they can monitor all outgoing traffic). Williamson's

technique [Wil02] is based on the observation that a corrupted computer will connect to as many computers as possible, while an uncorrupted computer makes repeated connection to recently accessed computers with higher probability. Since this technique provides an automatic response implemented through a delay queue and a series of time-outs, it does not affect the normal traffic. Similar techniques have been already marketed by Okena and Entercept.

7.2.3 Unused Block of IP Addresses

Eliciting erratic behavior (and extracting worm signatures) by combining wide address space monitors and host-based honeypot tools have already been studied. By using these techniques, researchers have successfully characterized and classified the traffic observed at unused blocks [Moo03, ZGT03a]. As noted in [CBM04] one interesting feature of the plots presented in that analysis were the differences in magnitude and composition of traffic between the different blocks. One approach for obtaining representative data is to increase the number and size of unused address blocks [Moo03]. To better understand how observed traffic is affected by sensor placement, data from Internet Monitor Sensor (IMS) is used in [CBM04] to present evidence that distributed unused address blocks observe significantly different traffic patterns. Unlike the approach in [Moo03] where a large unused address block is used

to obtain characterization of the global Internet traffic, the IMS utilizes a distributed collection of blackhole sensors. These sensors are deployed in networks belonging to service providers, large enterprises, and academic institutions representing a diverse sample of the IPv4 address space. The results of the analysis is that these distributed address blocks observe dramatically different traffic patterns.

7.3 Detection via Distributed Blackholes

Recently developed approaches for automatic extraction of worm signatures include detection through honeypots, virtual honeypots, honeynets, and blackholes (also known as network telescopes, darknets). A *honeypot* is a closely monitored network–decoy that can distract adversaries from more valuable machines on a network, provide early warning about new attack and exploitation trends, or can allow in–depth examination of adversaries during and after exploitation of the honeypot. While deploying a physical honeypot is often time intensive and expensive, *virtual honeypots* [Pro04] simulate virtual computer systems at the network level without the necessary hardware requirements. The simulated computer systems appear to run on unallocated network addresses. To deceive network fingerprinting tools, virtual honeypots simulates the networking stack of different operating systems and can provide arbitrary routing topologies and services for an arbitrary number of virtual systems. Unlike

honeypots and virtual honeypots, a *honeynet* [LLO03] is an entire network of systems that runs real applications and services. An attacker can interact with operating systems and execute tools on what appears to be a legitimate production network. In a network telescope, a portion of unused address space is globally announced and routed to a collection infrastructure that records incoming and/or outgoing packets. All captured activities are assumed to be unauthorized or malicious as any connection initiated inbound or outbound to these four systems is most likely a result of misconfiguration, or scanning from worms and other network probing. The signatures obtained from traffic analysis, in turn, can be used to devise agents that block attacks into real systems.

Our detection mechanism, *Detection via Distributed Blackholes* (*DDBH*), (1) belongs to the group of threshold-based algorithms, (2) operates via network traffic monitoring, and (3) uses distributed collection of unused address blocks (known as blackholes). The last characteristic renders DDBH deployable via (virtual) honeypots or honeynets. Moreover, DDBH provides the basis for coordinated defense by using only locally available information.

The DDBH architecture is a set of heterogeneous blackhole sensors, aggregators, and responders. Each blackhole sensor monitors a dedicated range of unused IP addresses. For each packet sent to the blackhole, the sensor records the source IP address, destination IP address, and the destination port. Because there are no

legitimate hosts in an unused address block, the traffic must be a result of poor routing management or scanning/probing activity. Each blackhole sensor is responsible for gathering and storing data, performing queries on its local storage, and generating alerts that are sent to the aggregators. The blackhole sensor looks for erratic activities such as: horizontal scan, vertical scan, or coordinated scans whose characterization has already been studied [VVK98, VCI99]. *Aggregators* communicate with the sensors to gather information about the global characteristics of the propagation and plan further actions, *e.g.*, alarming certain responders. Finally, *responders*, through two–way communication with analyzers and aggregators, initiate a pre–specified control strategy. The DDBH architecture is presented in Figure 7.1, below.

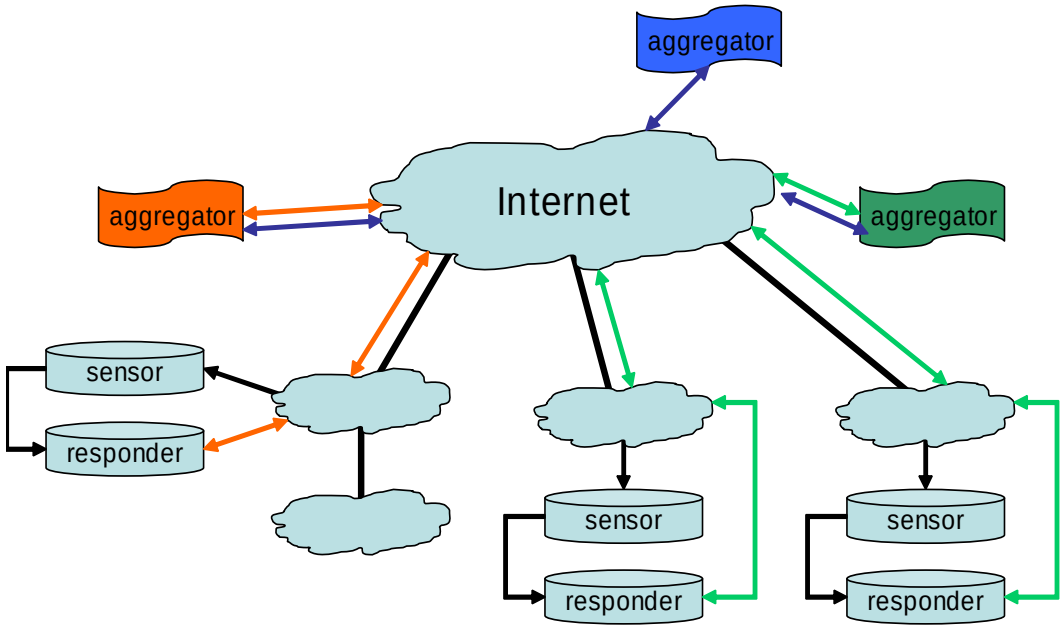


Figure 7.1: DDBH Algorithm at Aggregator

Let τ denote the threshold for the number of unused IP addresses on which scanning attempts are detected and t_d be the time when the threshold is exceeded. Each *aggregator* keeps a list of pairs (*destination address*, *source address*). When a blackhole sensor detects scanning activity for a particular destination address, it sends the pair of "infected" blackhole address and infectious (source) address to the aggregator. When the number of blackhole destination address, on which scanning was attempted, exceeds the threshold, the aggregator activates the *responders* responsible for handling the associated source addresses. The responders, in turn, through communication with the blackhole sensors and aggregators initiate a pre–specified control strategy. A responder that has been activated updates its actions based on the data from the corresponding sensor and aggregator. Therefore, a particular control strategy launched by a given responder can be terminated and redirected to another part of the network. The DDBH algorithm is formally described in Figure 7.2.

7.4 Model of DDBH with Contact-tracing

Contact tracing is a form of targeted control, where applying a chosen control mechanism is focused on the potential next-generation cases. *Spatially-explicit contact-tracing* is an epidemiological control strategy (see [TH03] and references therein) that has not yet been investigated in the context of worm quarantining. As contact

Algorithm *Detection via Distributed Blackholes*

Input:

$L_\emptyset$, list of pairs (infected $\emptyset$-node, infectious (regular) node neighbor)
ϑ, number of elements in $L_\emptyset$

```
1: if pair (∅, i) received from a sensor/child-aggregator then
2:     ϑ ← ϑ + 1
3: end if
4: if ϑ > τ then
5:     ϑ ← 0
6:     empty L_∅
7:     start control strategy at responders responsible for L_∅
8: end if
```

Figure 7.2: Detection via Distributed Blackholes algorithm

tracing is fundamentally linked to the network of potential transmission routes, classical epidemiological models are not suitable for its analysis. This control strategy looks for *patterns of infectious nodes* (once they have been identified), and hence utilizes the network structure associated with transmission. Contact tracing is modeled by investigating a proportion of the neighbors of an identified infectious node—this portion is referred to as *tracing efficiency*. Existing contact-tracing models [TH03, EK03] truncate the chains of transmission at adjacent nodes, *i.e.*, secondary cases. Here, we present a model of contact tracing that follows the entire chain of transmission.

We formalize the DDBH detection mechanism as a modification of the Susceptible-Infectious model employing information about the network structure: Nodes are divided into two groups—*regular nodes*, representing used IP addresses (hosts), and

$\emptyset$-*nodes*, modeling unused IP addresses (blackholes). Before the time moment t_d, when the worm is detected, a regular node can be either susceptible or *infectious*, while a $\emptyset$-node can be either susceptible or *infected*. Note that a $\emptyset$-node is never infectious, and thus, it does not facilitate the propagation.

Contact-tracing is incorporated into the detection framework via the Susceptible-Infectious-Traced-Removed model: at any time moment after t_d, a regular node can be in one of the four states—susceptible, infectious, traced, and removed, whereas a $\emptyset$-node can be susceptible, infected, or traced. When an infectious (regular) node moves in the traced state, it can no longer propagate the worm. Finally, if the number of infected $\emptyset$-nodes exceeds τ, all infected ($\emptyset$-node) enter the traced state and initiate contact-tracing. Therefore, the model allows for studying not only the effect of the distributed placement of blackholes but also coordination in the wake of an attack by only using local information.

Let β denote the rate at which a susceptible node is infected from an adjacent infectious node, μ be the rate at which nodes are traced, and γ be the rate at which nodes move out of the traced state (*i.e.*, are removed). Let for any time moment t, $[S]$ denote the number of susceptible regular nodes, $[I]$, the number of infectious nodes, $[T]$, the number of traced regular nodes, $[R]$, the number of removed nodes, $\left[S^{\emptyset}\right]$, the number of susceptible $\emptyset$-nodes, $\left[I^{\emptyset}\right]$, the number of infected nodes, and $\left[T^{\emptyset}\right]$ be the number of traced $\emptyset$-nodes. Furthermore, let $[XY]$ denote the number of pairs where

one node is in state X and the other is in state Y, and $[XYZ]$ be the number of triples in which node in state Y has two neighbors in states X and Z.

For simplicity, let G be a $\overline{d}$-regular graph on a set of regular and $\emptyset$-nodes. Following the approach in [Ran99], we derive the following set of equations for the dynamics of the defense system before t_d:

$$\begin{aligned}
\frac{d\,[S]}{dt} &= -\beta\,[SI]\,, \\
\frac{d\,[I]}{dt} &= \beta\,[SI]\,, \\
\frac{d\left[S^{\emptyset}\right]}{dt} &= -\beta\left[S^{\emptyset}I\right], \\
\frac{d\left[I^{\emptyset}\right]}{dt} &= \beta\left[S^{\emptyset}I\right],
\end{aligned} \tag{7.4.1}$$

where initially $[I] > 0$, $\left[I^{\emptyset}\right] = 0$. At any time moment before t_d, $[S] + [I] = n$, $\left[S^{\emptyset}\right] + \left[I^{\emptyset}\right] = n^{\emptyset}$. The first two equations in model (7.4.1) describe the propagation of the worm on the regular nodes, while the last two equations model the worm scanning captured by the $\emptyset$-nodes. Similarly, the dynamics of contact-tracing, initiated at time moment t_d, is given by:

$$
\begin{aligned}
\frac{d\,[S]}{dt} &= -\beta\,[SI]\,, \\
\frac{d\,[I]}{dt} &= \beta\,[SI] - \mu\left([TI] - \left[T^{\emptyset}I\right]\right), \\
\frac{d\,[T]}{dt} &= \mu\,[TI] - \gamma\,[T]\,, \\
\frac{d\,[R]}{dt} &= \gamma\,[T]\,, \\
\frac{d\left[S^{\emptyset}\right]}{dt} &= -\beta\left[S^{\emptyset}I\right], \\
\frac{d\left[I^{\emptyset}\right]}{dt} &= \beta\left[S^{\emptyset}I\right],
\end{aligned}
\tag{7.4.2}
$$

where, initially, $\left[T^{\emptyset}\right]$ equals to the value of $\left[I^{\emptyset}\right]$ when $\vartheta = \tau_{ct}$, and $\left[I^{\emptyset}\right] = 0$. At any time moment after t_d, $[S] + [I] + [T] + [R] = n$, $\left[S^{\emptyset}\right] + \left[I^{\emptyset}\right] + \left[T^{\emptyset}\right] = n^{\emptyset}$.

The dynamics of the pairs before the time moment t_d is given below:

$$
\begin{aligned}
\frac{d\,[SS]}{dt} &= -2\beta\,[SSI]\,, \\
\frac{d\,[SI]}{dt} &= \beta\,([SSI] - [ISI] - [SI])\,, \\
\frac{d\,[II]}{dt} &= 2\beta\,([ISI] + [SI])\,, \\
\frac{d\,\left[S^{\emptyset}S\right]}{dt} &= -\beta\left(\left[S^{\emptyset}SI\right] + \left[IS^{\emptyset}S\right]\right), \\
\frac{d\,\left[S^{\emptyset}I\right]}{dt} &= \beta\left(\left[S^{\emptyset}SI\right] - \left[IS^{\emptyset}I\right] - \left[S^{\emptyset}I\right]\right), \\
\frac{d\,\left[I^{\emptyset}S\right]}{dt} &= \beta\left(\left[IS^{\emptyset}S\right] - \left[I^{\emptyset}SI\right]\right), \\
\frac{d\,\left[I^{\emptyset}I\right]}{dt} &= \beta\left(\left[S^{\emptyset}I\right] + \left[IS^{\emptyset}I\right] + \left[I^{\emptyset}SI\right]\right),
\end{aligned}
\tag{7.4.3}
$$

where at any time moment, the sum of all possible pairs equals twice the edges, *i.e.*, $n\overline{d}$. The pair dynamics of contact-tracing is given by:

$$
\begin{aligned}
\frac{d\,[SS]}{dt} &= -2\beta\,[SSI]\,, \\
\frac{d\,[SI]}{dt} &= \beta\,([SSI] - [ISI] - [SI]) - \mu\,[SIT]\,, \\
\frac{d\,[ST]}{dt} &= \mu\,[SIT] - \gamma\,[ST]\,, \\
\frac{d\,[SR]}{dt} &= \gamma\,[ST]\,, \\
\frac{d\,[II]}{dt} &= 2\beta\,([ISI] + [SI]) - 2\mu\,[IIT]\,, \\
\frac{d\,[IT]}{dt} &= \mu\,[IIT] - \gamma\,[IT]\,, \\
\frac{d\,[IR]}{dt} &= \gamma\,[IT]\,, \\
\frac{d\,[TT]}{dt} &= 2\mu\,[TIT] - 2\gamma\,[TT]\,, \\
\frac{d\,[TR]}{dt} &= \gamma\,[TT]\,, \\
\frac{d\,[RR]}{dt} &= 2\gamma\,[TR]\,,
\end{aligned}
\tag{7.4.4}
$$

while for the interaction among regular and $\emptyset$-nodes one can obtain:

$$
\begin{aligned}
\frac{d\left[S^{\emptyset}S\right]}{dt} &= -\beta\left(\left[S^{\emptyset}S\right] + \left[IS^{\emptyset}S\right]\right), \\
\frac{d\left[S^{\emptyset}I\right]}{dt} &= \beta\left(\left[S^{\emptyset}SI\right] - \left[IS^{\emptyset}I\right] - \left[S^{\emptyset}I\right]\right) - \mu\left[S^{\emptyset}IT\right], \\
\frac{d\left[S^{\emptyset}T\right]}{dt} &= \mu\left[S^{\emptyset}IT\right] - \gamma\left[S^{\emptyset}T\right], \\
\frac{d\left[S^{\emptyset}R\right]}{dt} &= \gamma\left[S^{\emptyset}T\right], \\
\frac{d\left[I^{\emptyset}S\right]}{dt} &= \beta\left(\left[IS^{\emptyset}S\right] - \left[I^{\emptyset}SI\right]\right), \\
\frac{d\left[I^{\emptyset}I\right]}{dt} &= \beta\left(\left[S^{\emptyset}I\right] + \left[IS^{\emptyset}I\right] + \left[I^{\emptyset}SI\right]\right) - \mu\left[I^{\emptyset}IT\right], \\
\frac{d\left[I^{\emptyset}T\right]}{dt} &= \mu\left[I^{\emptyset}I\right] - \gamma\left[I^{\emptyset}T\right], \\
\frac{d\left[I^{\emptyset}R\right]}{dt} &= \gamma\left[I^{\emptyset}T\right], \\
\frac{d\left[T^{\emptyset}S\right]}{dt} &= -\beta\left[T^{\emptyset}SI\right], \\
\frac{d\left[T^{\emptyset}I\right]}{dt} &= -\beta\left[T^{\emptyset}SI\right] - \mu\left[T^{\emptyset}I\right], \\
\frac{d\left[T^{\emptyset}T\right]}{dt} &= \mu\left[T^{\emptyset}I\right] - \gamma\left[T^{\emptyset}T\right], \\
\frac{d\left[T^{\emptyset}R\right]}{dt} &= \gamma\left[T^{\emptyset}T\right].
\end{aligned}
\tag{7.4.5}
$$

To approximate the third moments $[XYZ]$, one can use the estimate derived in [Ran99]:

$$[XYZ] = \frac{\overline{d}-1}{\overline{d}} \frac{[XY]\,[YZ]}{[Y]}. \tag{7.4.6}$$

In order to study the effects of distributed placement of $\emptyset$-nodes, we extend the model given by (7.4.1) - (7.4.3) to include the full network structure, according to the model presented in Chapter 5:

$$\begin{aligned}
\frac{d\,[I_a]}{dt} &= \beta \sum_{k\in\Lambda_a} [S_a I_k], \\
\frac{d\,[S_a]}{dt} &= -\beta \sum_{k\,\in\,\Lambda_a} [S_a I_k], \\
\frac{d\left[I_a^{\emptyset}\right]}{dt} &= \beta \sum_{k\in\Lambda_a} \left[S_a^{\emptyset} I_k\right], \\
\frac{d\left[S_a^{\emptyset}\right]}{dt} &= -\beta \sum_{k\in\Lambda_a} \left[S_a^{\emptyset} I_k\right].
\end{aligned} \tag{7.4.7}$$

The dynamics of the pairs of regular nodes is then given by:

$$
\begin{aligned}
\frac{d\,[S_aS_b]}{dt} &= -\beta\left(\sum_{k\in\Lambda_a}[I_kS_aS_b] + \sum_{k\,\in\,\Lambda_b}[S_aS_bI_k]\right),\\
\frac{d\,[S_aI_b]}{dt} &= -\beta\left(\sum_{k\in\Lambda_a}[I_kS_aI_b] - \sum_{k\,\in\,\Lambda_b}[S_aS_bI_k] + [S_aI_b]\right),\\
\frac{d\,[I_aS_b]}{dt} &= -\beta\left(\sum_{k\in\Lambda_b}[I_aS_bI_k] - \sum_{k\,\in\,\Lambda_a}[I_kS_aS_b] + [I_aS_b]\right),\\
\frac{d\,[I_aI_b]}{dt} &= \beta\left([S_aI_b] + [I_aS_b] + \sum_{k\in\Lambda_a}[I_kS_aI_b] + \sum_{k\in\Lambda_b}[I_aS_bI_k]\right),
\end{aligned}
\tag{7.4.8}
$$

while the interaction between the regular and $\emptyset$-nodes can be described by:

$$
\begin{aligned}
\frac{d\left[S_a^{\emptyset} S_b\right]}{dt} &= -\beta\left(\sum_{k\in\Lambda_a}\left[I_k S_a^{\emptyset} S_b\right] + \sum_{k\,\in\,\Lambda_b}\left[S_a^{\emptyset} S_b I_k\right]\right) \\
\frac{d\left[S_a^{\emptyset} I_b\right]}{dt} &= \beta\left(\sum_{k\,\in\,\Lambda_b}\left[S_a^{\emptyset} S_b I_k\right] - \sum_{k\in\Lambda_a}\left[I_k S_a^{\emptyset} I_b\right] - \left[S_a^{\emptyset} I_b\right]\right), \\
\frac{d\left[I_a^{\emptyset} S_b\right]}{dt} &= \beta\left(\sum_{k\in\Lambda_a}\left[I_k S_a^{\emptyset} S_b\right] - \sum_{k\in\Lambda_b}\left[I_a^{\emptyset} S_b I_k\right]\right), \\
\frac{d\left[I_a^{\emptyset} I_b\right]}{dt} &= \beta\left(\left[S_a^{\emptyset} I_b\right] + \sum_{k\in\Lambda_a}\left[I_k S_a^{\emptyset} I_b\right] + \sum_{k\in\Lambda_b}\left[I_a^{\emptyset} S_b I_k\right]\right).
\end{aligned}
\tag{7.4.9}
$$

To solve the model, observe that the approximation given by (7.4.6) could be extended to:

$$
[X_a Y_b Z_c] = \frac{b-1}{b}\frac{[X_a Y_b]\,[Y_b Z_c]}{[Y_b]}. \tag{7.4.10}
$$

Similar arguments can be used to extend model (7.4.4) - (7.4.5).

7.5 Analysis of DDBH with Contact-Tracing

In this section we analyze the model presented in Section 7.4 above by means of the individual-based simulation described in Section 5.4. First, we need to choose where to place the $\emptyset$-nodes and how many of them to place. We note that the study in [MVS03] concluded that almost all network paths should be monitored in order to effectively control the worm propagation.

Given a graph G, let $V'(G)$ be the minimum vertex cover of G. Here, for every node u of highest degree from $V'(G)$, a new *emptyset*-node with same neighbors as u is added to G. Thus, the $\emptyset$-nodes are added by copying the first $n_\emptyset$ nodes of highest degree in $V'(G)$.

Since the problem of finding a minimum vertex cover of G is NP-hard, we use the heuristic presented in Figure 7.3 to find an approximation. The heuristic is a greed algorithm that at each step chooses a node (not in $V'(G)$) for inclusion in the vertex cover.

The size of the vertex covers for five Macroscopic Internet graphs are shown in Figure 7.4. Given a Macroscopic Internet graph G, the number of $\emptyset$-nodes added to G is 1% (respectively, 2%) of the order of G when $\emptyset$-nodes comprise 4% (respectively,

Algorithm *Vertex cover*

Input: G, graph

Output: $V'(G)$, vertex cover of G

$V'(G) \leftarrow \emptyset$
while there $|V(G)| > 0$ **do**
  $v \leftarrow$ node of highest degree
  $V'(G) \leftarrow V'(G) \cup v$
  $G = G - v$
end while

Figure 7.3: Vertex cover heuristic

8%) of the highest-degree nodes in the vertex cover. These values allows comparison of our results with the outcome of previous studies, presented in Chapter 6.

Graph	Order	Size of Vertex Cover (VC)	4% of VC		8% of VC	
AS 08.11.1997	3015	588	24	0.78%	47	1.56%
AS 02.20.1998	4180	829	33	0.79%	66	1.59%
AS 02.07.1999	5357	964	39	0.72%	77	1.44%
AS 02.07.2000	7956	1266	51	0.64%	101	1.27%
AS 03.16.2001	10515	1640	66	0.62%	131	1.25%

Figure 7.4: Number of nodes in vertex cover of Macroscopic Internet graphs

Three sets of experiments were performed for each graph and each number of $\emptyset$-nodes added when $\tau = 1$, 3, or 6. An increase in the value of τ, renders the DDBH algorithm more sensitive to false positives, but at the same time, it increases the time

before the worm is detected. By varying the parameter τ, we were able to determine how the DDBH algorithm performs for the three levels of sensitivity to false alarms.

In all experiments, the parameter $\gamma = 0$, in order to model the case when the infectious nodes are only traced (*i.e.*, their traffic is blocked) without removing the worm. With the help of the individual-based simulation we were able to answer questions related to:

1. The time, t_d, necessary to detect the worm,

2. The number of infectious nodes when the worm is detected, and

3. The number of nodes at time t_e, when all infectious nodes are traced.

As in Section 6.5, the fraction of susceptible nodes at the end of propagation (at time moment t_e) is used as a measure of the effectiveness of contact-tracing. The results of the empirical study for the Macroscopic Internet graphs from 08.11.1997 appear in Figures 7.5. The results for the Macroscopic Internet graph from 02.07.2000 are shown in Figures 7.6.

To summarize the results:

1. With the increase of τ, (i) the time, t_d to detect the worm, (ii) the number of infectious nodes, $I(t_d)$ when the worm is detected increase, and (iii) the fraction of susceptible nodes $S(t_e)$ at the end of the propagation, increase.

AS 08.11.1997		τ = 1	4% of vertex cover monitored				
β	μ	average td	σ td	average % I(td)	σ % I(td)	te	% S(te)
1.8	3.6	0.217	0.187	0.184%	0.208%	6.529	59.575%
1.8	1.8	0.255	0.214	0.237%	0.326%	6.953	55.841%
1.8	0.4	0.241	0.233	0.228%	0.235%	20.042	33.272%
1.5	3.6	0.245	0.173	0.199%	0.229%	7.549	59.379%
1.5	1.8	0.298	0.237	0.222%	0.235%	8.566	60.101%
1.5	0.4	0.309	0.286	0.237%	0.229%	20.069	37.884%
0.9	3.6	0.481	0.398	0.166%	0.172%	10.118	64.993%
0.9	1.8	0.497	0.407	0.191%	0.290%	11.889	64.724%
0.9	0.4	0.473	0.389	0.162%	0.177%	20.488	48.144%

AS 08.11.1997		τ = 1	8% of vertex cover monitored				
β	μ	average td	σ td	average % I(td)	σ % I(td)	te	% S(te)
1.8	3.6	0.271	0.235	0.167%	0.190%	7.322	61.502%
1.8	1.8	0.225	0.179	0.170%	0.207%	7.593	59.835%
1.8	0.4	0.182	0.232	0.188%	0.244%	19.534	40.856%
1.5	3.6	0.321	0.296	0.154%	0.174%	8.562	63.481%
1.5	1.8	0.207	0.232	0.120%	0.198%	8.120	67.685%
1.5	0.4	0.265	0.197	0.192%	0.213%	20.477	43.207%
0.9	3.6	0.507	0.419	0.149%	0.177%	11.444	68.450%
0.9	1.8	0.595	0.606	0.188%	0.174%	15.550	57.307%
0.9	0.4	0.558	0.419	0.167%	0.197%	20.062	55.799%

(a)

2. With the decrease of μ, the fraction of susceptible nodes $S(t_e)$ at the end of the propagation (when all infectious nodes are traced) decreases.

3. DDBH algorithm could detect the worm when less than 4% of nodes are infected.

Moreover, the experimental analysis showed that by integrating contact-tracing within the DDBH architecture more than 40% of the susceptible nodes can be protected.

AS 08.11.1997		τ = 3	4% of vertex cover monitored				
β	μ	average td	σ td	average % I(td)	σ % I(td)	te	% S(te)
1.8	3.6	0.434	0.337	0.696%	0.469%	8.627	50.960%
1.8	1.8	0.464	0.355	0.813%	0.578%	7.741	53.877%
1.8	0.4	0.478	0.348	0.635%	0.438%	20.270	37.375%
1.5	3.6	0.464	0.271	0.778%	0.459%	9.937	47.914%
1.5	1.8	0.521	0.356	0.626%	0.436%	9.728	54.259%
1.5	0.4	0.542	0.412	0.657%	0.488%	21.204	39.752%
0.9	3.6	1.137	1.232	0.677%	0.468%	15.569	47.330%
0.9	1.8	0.774	0.754	0.673%	0.421%	17.018	51.922%
0.9	0.4	0.879	0.602	0.704%	0.477%	20.989	47.132%

AS 08.11.1997		τ = 3	8% of vertex cover monitored				
β	μ	average td	σ td	average % I(td)	σ % I(td)	te	% S(te)
1.8	3.6	0.534	0.450	0.654%	0.392%	9.614	53.167%
1.8	1.8	0.398	0.259	0.647%	0.335%	8.696	56.175%
1.8	0.4	0.539	0.381	0.647%	0.380%	21.263	40.919%
1.5	3.6	0.572	0.388	0.582%	0.389%	10.396	50.319%
1.5	1.8	0.534	0.566	0.602%	0.395%	10.836	56.783%
1.5	0.4	0.580	0.436	0.625%	0.416%	21.041	44.076%
0.9	3.6	0.890	0.671	0.632%	0.434%	16.991	49.492%
0.9	1.8	0.808	0.750	0.574%	0.330%	18.434	53.162%
0.9	0.4	0.929	0.826	0.540%	0.333%	21.935	51.374%

(b)

7.6 Summary

A prerequisite for a defense system is its ability to identify the presence of the worm on the network. First, we describe the existing worm detection mechanism, and point out their advantages and disadvantages. As obtaining attack-free traffic data is almost impossible and a large block of unused IP addresses does not give a clear picture for the global propagation of the worm, here we designed a distributed detection architecture

called *Detection via Distributed Blackholes* (*DDBH*). Our novel detection mechanism could be implemented via virtual honeypots or honeynets. Finally, a novel control strategy called contact-tracing is incorporated in the DDBH. Simulation results show that the worm can be detected with virtual honeypots on only 3% of the nodes. Moreover, the worm is detected when less than 4% of the nodes are infected. Contact-tracing integrated in the DDBH can protected more than 40% of the susceptible nodes.

AS 08.11.1997		τ = 6	4% of vertex cover monitored				
β	μ	average td	σ td	average % I(td)	σ % I(td)	te	% S(te)
1.8	3.6	0.476	0.318	1.554%	0.716%	8.997	51.788%
1.8	1.8	0.465	0.344	1.406%	0.693%	8.029	55.340%
1.8	0.4	0.428	0.361	1.407%	0.632%	20.064	39.015%
1.5	3.6	0.541	0.424	1.295%	0.675%	10.653	49.763%
1.5	1.8	0.648	0.470	1.455%	0.704%	10.144	55.389%
1.5	0.4	0.518	0.376	1.555%	0.725%	20.762	43.222%
0.9	3.6	0.960	0.674	1.321%	0.683%	16.405	47.955%
0.9	1.8	1.125	0.828	1.378%	0.725%	18.030	50.180%
0.9	0.4	1.094	0.778	1.373%	0.608%	22.049	50.519%

AS 08.11.1997		τ = 6	8% of vertex cover monitored				
β	μ	average td	σ td	average % I(td)	σ % I(td)	te	% S(te)
1.8	3.6	0.611	0.437	1.210%	0.617%	9.220	52.807%
1.8	1.8	0.416	0.283	1.274%	0.586%	8.964	57.960%
1.8	0.4	0.473	0.309	1.244%	0.513%	20.361	42.479%
1.5	3.6	0.586	0.391	1.115%	0.512%	11.007	52.076%
1.5	1.8	0.480	0.308	1.073%	0.468%	11.696	57.751%
1.5	0.4	0.546	0.429	1.390%	0.661%	20.243	45.499%
0.9	3.6	0.802	0.517	1.292%	0.511%	18.371	50.018%
0.9	1.8	1.048	1.023	1.171%	0.493%	19.820	53.526%
0.9	0.4	1.073	0.776	1.256%	0.611%	21.627	53.546%

(c)

Figure 7.5: Statistical analysis of contact-tracing integrated with DDBH on Macroscopic Internet graph from 08.11.1997, (a) $\tau = 1$, (b) $\tau = 3$, (c) $\tau = 6$

AS 02.07.2000		τ = 3	4% of vertex cover monitored				
β	μ	average t_d	σ t_d	average % I(t_d)	σ % I(t_d)	t_e	% S(t_e)
1.8	3.6	0.203	0.150	0.367%	0.284%	8.275	53.807%
1.8	1.8	0.220	0.178	0.367%	0.256%	7.936	57.920%
1.8	0.4	0.204	0.164	0.364%	0.229%	22.548	39.575%
1.5	3.6	0.287	0.268	0.417%	0.299%	9.123	55.578%
1.5	1.8	0.228	0.211	0.396%	0.222%	10.448	56.610%
1.5	0.4	0.185	0.163	0.475%	0.283%	23.441	42.832%
0.9	3.6	0.342	0.303	0.417%	0.271%	15.240	45.035%
0.9	1.8	0.380	0.292	0.411%	0.272%	16.872	52.318%
0.9	0.4	0.352	0.265	0.422%	0.246%	23.334	50.498%

(a)

AS 02.07.2000		τ = 6	4% of vertex cover monitored				
β	μ	average t_d	σ t_d	average % I(t_d)	σ % I(t_d)	t_e	% S(t_e)
1.8	3.6	0.224	0.211	0.783%	0.350%	8.759	52.572%
1.8	1.8	0.189	0.125	0.829%	0.365%	8.726	58.369%
1.8	0.4	0.220	0.191	0.765%	0.347%	23.368	41.434%
1.5	3.6	0.230	0.172	0.796%	0.321%	9.786	50.130%
1.5	1.8	0.241	0.191	0.780%	0.410%	10.697	57.497%
1.5	0.4	0.229	0.207	0.752%	0.340%	23.816	44.956%
0.9	3.6	0.368	0.273	0.875%	0.366%	16.346	45.323%
0.9	1.8	0.380	0.285	0.782%	0.355%	17.837	52.755%
0.9	0.4	0.479	0.388	0.801%	0.366%	23.436	52.144%

(b)

Figure 7.6: Statistical analysis of contact-tracing integrated with DDBH on Macroscopic Internet graph from 02.07.2000, (a) $\tau = 3$, (b) $\tau = 6$

LIST OF REFERENCES

[AB02] R. Albert and A. Barabasi. "Statistical Mechanics of Complex Networks." *Reviews of Modern Physics*, **74**:47–97, 2002.

[ACL00] W. Aiello, F. Chung, and L. Lu. "A Random Graph Model of Massive Graphs." In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pp. 171–180, 2000.

[AF84] M. Aigner and M. Fromme. "A Game of Cops and Robbers." *Discrete Applied Mathematics*, **8**:1–12, 1984.

[AJB99] R. Albert, H. Jeong, and A. L. Barabasi. "Diameter of the World Wide Web." *Nature*, **401**(130-131), 1999.

[AKL79] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovsz, and C. Rackoff. "Random Walks, Universal Traversal Sequences, and the Complexity of Maze Problems." In *Proceedings of the IEEE Symposium on Foundation of Computer Science*, pp. 218–223, 1979.

[AM92] R. M. Anderson and R. M. May. *Infectious Diseases in Humans.* Oxford University Press, 1992.

[AM03] W. Allen and G. Marin. "Detecting New Denial–of–Service Attacks Without a Traffic Template." In *Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, 2003.

[And86] T. Andreae. "On a Pursuit Game Played on Graphs for which Minor is Excluded." *Journal of Combinatorial Theory Series B*, **41**:37–47, 1986.

[ARS02] M. Adler, H. Racke, N. Sivadasan, C. Sohler, and B. Vocking. "Randomized Pursuit–Evasion in Graphs." *Lecture Notes in Computer Science*, **2380**:901–912, 2002.

[Att00] A. Attoui. *Real–Time and Multi–Agent Systems.* Springer–Verlag, 2000.

[BA99] A. Barabasi and R. Albert. "Emergence of Scaling in Random Networks." *Science*, **286**:509–512, 1999.

[Bau02] C. T. Bauch. "A Versatile ODE Approximation to a Network Model for the Spread of Sexually Transmitted Diseases." *Journal of Mathematical Biology*, **45**(375-395), 2002.

[BB03] V. H. Berk and G. Bakos. "Designing a Framework for Active Worm Detection on Global Networks." In *Proceedings of the IEEE International Workshop on Information Assurance*, 2003.

[BGB02] V. H. Berk, R. S. Gray, and G. Bakos. "Using Sensor Networks and Data Fusion for Early Detection of Active Worms." In *Proceedings of the SPIE conference on Sensors, and Command, Control, Communications and Intelligence*, 2002.

[BKM00] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. "Graph Structure in the Web." *Computer Networks*, **33**:309–315, 2000.

[BO98] T. Basar and G.J. Olsder. *Dynamic Non–cooperative Game Theory.* SIAM, 1998.

[BO04] P.G. Buckley and D. Osthus. "Popularity based Random Graph Model Leading to a Scale–free Degree Sequence." *Discrete Mathematics*, **282**:53–68, 2004.

[Bol85] B. Bollobas. *Random Graphs.* Academic Press Inc., 1985.

[Bol03] B. Bollobas. "Mathematical Results on Scale–free Graphs." *preprint*, 2003.

[BPV03] M. Boguna, R. Pastor-Satorras, and A. Vespignani. *Lecture Notes in Physics*, volume 625, chapter Epidemic Spreading in Complex Networks with Degree Correlations, pp. 127–147. 2003.

[BR00] D. Bruschi and E. Rosti. "Disarming Offense to Facilitate Defense." In *Proceedings of the New Security Paradigms Workshop*, pp. 69–75, 2000.

[BR04] B. Bollobas and O. M. Riordan. "The Diameter of a Scale–free Random Graph." *Combinatorica*, **24**(1):5–34, 2004.

[Bra97] J. Bradshaw. *Software Agents.* The MIT Press, 1997.

[BS91] D. Bienstock and P. Seymour. "Monotonicity in Graph Searching." *Journal of Algorithms*, **12**:239–245, 1991.

[BW00] A. Barrat and M. Weigt. "On the Properties of Small–World Network Models." *European Physical Journal B*, **13**:547–560, 2000.

[CBM04] E. Cooke, M. Bailey, Z. Morley Mao, and D. McPherson. "Toward Understanding Distributed Blackhole Placement." In *Proceedings of the ACM Workshop on Rapid Malcode*, pp. 54–54, 2004.

[Cen01a] CERT Coordination Center. "Advisory CA-2001-19. Available at: www.cert.org/advisories/CA-2001-19.html." 2001.

[Cen01b] CERT Coordination Center. "Advisory CA-2001-26. Available at: www.cert.org/advisories/CA-2001-26.html." 2001.

[Cen03] CERT Coordination Center. "Advisory CA-2001-19. Available at: www.cert.org/advisories/CA-2003-20.html." 2003.

[CF03] C. Cooper and A. Frieze. "A General Model of Web Graphs." *Random Structures and Algorithms*, **22**(3):311–335, 2003.

[CGK03] Z. Chen, L. Gao, and K. Kwiat. "Modeling the Spread of Active Worms." In *Proceedings of the IEEE INFOCOM*, 2003.

[CL01] F. Chung and L. Lu. "The Diameter of Random Sparse Graphs." *Advances in Applied Mathematics*, **26**, 2001.

[CLV03] F. Chung, L. Lu, and V. Vu. "Spectra of Random Graphs with Given Expected Degrees." In *Proceedings of the National Academy of Sciences of the United States of America*, volume 100, pp. 6313–6318, 2003.

[COP01] J. Cowie, A. T. Ogielski, B. J. Premore, and Y. Yuan. "Global Routing Instabilities Triggered by Code Red II and Nimda. Available at: www.renesys.com." 2001.

[CR04] S. Chen and S. Ranka. "An Internet–worm Early Warning System." In *Proceedings of the IEEE GLOBECOM 2004 - Security and Network Management*, volume 4, pp. 2261–2265, 2004.

[CRM00] J. Cabrera, B. Ravichandran, and R. Mehra. "Statistical Traffic Modeling for Network Intrusion Detection." In *Proceedings of the 8th IEEE Symposium on Modeling, Analysis, and Simulation of Computers and Telecommunications*, 2000.

[CT04] S. Chen and Y. Tang. "Slowing Down Internet Worms." In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS)*, pp. 312–319, 2004.

[DF02] S. N. Dorogovtsev and J. F. F.Mendes. "Evolution on Networks." *Advances in Physics*, **51**, 2002.

[DFS00] S. N. Dorogovtsev, J. F. F.Mendes, and A. N. Samukhin. "Structure of Growing Networks with Preferential Linking." *Physical Review Letters*, **85**, 2000.

[DKM02] S. Dill, R. Kumar, K. S. Mccurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins. "Self–similarity in the Web." *ACM Transactions on Internet Technology*, **2**(3):205–223, 2002.

[DKT97] N. Dendris, L. Kirousis, and D. Thilikos. "Fugitive–search Games on Graphs and Related Parameters." *Theoretical Computer Science*, pp. 233–254, 1997.

[EK02a] K. T. D. Earnes and M. J. Keeling. "Modeling Dynamic and Network Heterogeneities in the Spread of Sexually Transmitted Diseases." In *Proceedings of the National Academy of Sciences*, volume 99, pp. 13330–13335, 2002.

[EK02b] V. M. Eguiluz and K. Klemm. "Epidemic Threshold in Structured Scale–free Networks." *Physics Review Letters*, **89**(108701), 2002.

[EK03] K. T. D. Eames and M. J. Keeling. "Contact Tracing and Disease Control." In *Proceedings of the Royal Society of London B*, volume 270, 2003.

[ER59] P. Erdos and A. Renyi. "On Random Graphs I." *Publicationes Mathematicae Debrecen*, **5**:290–297, 1959.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos. "On Power–Law Relationships of the Internet Topology." In *Proceedings of SIGCOMM*, 1999.

[Fra87] P. Frankl. "Cops and Robbers in Graphs with Large Girth and Cayley Graphs." *Discrete Applied Mathematics*, **17**:301–305, 1987.

[Gil56] N. Gilbert. "Enumeration of Labeled Graphs." *Canadian Journal of Mathematics*, **8**:405–411, 1956.

[GJ99] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman, 1999.

[GLL99] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. "A Visibility–based Pursuit–evasion problem." *International Journal of Computational Geometry and Applications*, **9**(4/5):471, 1999.

[GR95] A. S. Goldstein and E. M. Reingold. "The Complexity of Pursuit–evasion on a Graph." *Theoretical Computer Science*, **143**:93–112, 1995.

[GSQ05] G. Gu, M. Sharif, X. Qin, D. Dragon, W. Lee, and G. Riley. "Worm Detection, Early Warning and Response, Based on Local Victim Infromation." *preprint*, 2005.

[GT00] R. Govindan and H. Tangmunarunkit. "Heuristics for Internet Map Discovery." In *Proceedings of the IEEE INFOCOM*, 2000.

[Het00] H. W. Hethcote. "Mathematics of Infectious Diseases." *SIAM Review*, **42**(4):599–653, 2000.

[HM04] G. Hahn and G. MacGillivray. "A Characterization of k–cop–win Graphs and Digraphs." preprint, 2004.

[IKK04] V. Isler, S. Kannan, and S. Khanna. "Randomized Pursuit–evasion with Limited Visibility." In *Proceedings of the 15th Annual ACM-SIAM symposium on Discrete Algorithms*, 2004.

[Ins04] Computer Security Institute. "Ninth Annual Computer Crime and Security Survey. Available at: i.cmpnet.com." 2004.

[Kee99] M. J. Keeling. "The Effects of Local Spatial Structure on Epidemiological Invasions." In *Proceedings of the Royal Society of London B*, volume 266, pp. 859–867, 1999.

[Kep94] J. O. Kephart. *Artificial Life III*, chapter How Topology Affects Population Dynamics. Addison–Wesley, 1994.

[KP86] L. Kirousis and C. Papadimitriou. "Searching and Pebbling." *Theoretical Computer Science*, **47**:205–218, 1986.

[KR01] P. L. Krapivsky and S. Redner. "Organization of Growing Random Networks." *Physical Review E*, **066123**, 2001.

[KRD04] K. Kim, S. Radhakrishnan, and S. K. Dhall. "Measurement and Analysis of Worm Propagation on Internet Network Topology." In *Proceedings of the IEEE Conference on Computer Communications and Networks*, 2004.

[KRR99] R. Kumar, P. Raghavan, S. Rajagopalan, and A.S. Tomkins. "Trawling the Web for Emerging Cyber–communities." *Computer Networks*, **31**:1481–1493, 1999.

[KRR00] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, and E. Upfal. "Random Graph Models for the Web Graph." In *Proceedings of the 41st FOCS*, pp. 57–65, 2000.

[KW91] J. O. Kephart and S. R. White. "Directed–graph Epidemiological Models of Computer Viruses." In *Proceedings of the IEEE Symposium on Security and Privacy*, p. 343, 1991.

[LaP93] A. LaPaugh. "Recontamination Does not Help to Search a Graph." *Journal of the ACM*, **40**:224–245, 1993.

[LLO03] J. Levine, R. LaBella, H. Owen, D. Contis, and B. Culver. "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks." In *Proceedings of the 2003 IEEE Workshop on Information Assurance*, 2003.

[LN04] M. Liljenstam and D. M. Nicol. "Comparing Passive and Active Worm Defenses." In *Proceedings of the First International Conference on the Quantitative Evaluation of Systems (QEST)*, pp. 18–27, 2004.

[LNB03] M. D. Liljenstam, M. Nicol, V. H. Berk, and R. S. Gray. "Simulating Realistic Network Worm Traffc for Worm Warning System Design and Testing." In *Proceedings of the ACM Workshop on Rapid Malcode*, 2003.

[LTW94] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. "On the Self–similar Nature of Ethernet Traffic." *IEEE-ACM Transactions on Networking*, **2**(1), 1994.

[MHG88] N. Megiddo, S. Hakimi, M. Garey, D. Johnson, and C. Papadimitriou. "The Complexity of Searching a Graph." *Journal of the ACM*, **35**:18–44, 1988.

[Mit04] M. Mitzenmacher. "A Brief History of Generative Models for Power Law and Lognormal Distributions." *Internet Mathematics*, **1**(2):226–251, 2004.

[MM87] M. Maamoun and H. Meyniel. "On a Game of Policemen and Robber." *Discrete Applied Mathematics*, **17**:301–305, 1987.

[Moo03] D. Moore. "Network Telescopes. Available at: www.caida.org." 2003.

[MP02] M. Mihail and C. Papadimitriou. "On the Eigenvalue Power Law." *preprint*, 2002.

[MPS03] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. "Inside the Slammer Worm. Available at: www.computer.org/security." 2003.

[MR95] M. Molloy and B. A. Reed. "A Critical Point for Random Graphs with a Given Degree Sequence." *Random Structures and Algorithms*, **6**(2/3):161–180, 95.

[MSK01] S. McClure, J. Scambray, and G. Kurtz. *Hacking Exposed*, chapter Scanning. McGraw–Hill, 2001.

[MVS03] D. Moore, G. M. Voelker, C. Shannon, and S. Savage. "Internet Quarantine: Requirements for Containing Self–Propagating Code." In *Proceedings of the IEEE INFOCOM*, 2003.

[MZ01] W. Jia M. Li and W. Zhao. "Decision Analysis of Network–based Intrusion Detection System for Denial–of–Service Attacks." In *Proceedings of the IEEE Conference on Info-tech and Info-net*, 2001.

[Nac99] C. Nachenberg. "Computer Parasitology." In *Proceedings of the 9th International Virus Bulletin Conference*, pp. 7–26, 1999.

[NAW01] J. Nazario, J. Anderson, R. Wash, and C. Connelly. "The Future of Internet Worms. Available at: www.crimelabs.net." 2001.

[New89] M. E. J. Newman. "Assortative Mixing in Networks." *Physical Review Letters*, **208701**, 89.

[NL04] D. M. Nicol and M. Liljenstam. "Models of Active Worm Defenses." In *Proccedings of the IPSI Studenica Conference*, 2004.

[NN93] S. Neufeld and R. Nowakowski. "A Vertex–to-Vertex Pursuit Game Played on Disjoint Sets of Edges." In N. Sauer, editor, *Proceedings of the NATO Advanced Study Institute on Finite and Infinite Combinatorics in Sets and Logic*, pp. 299–312, 1993.

[NN98] S. Neufeld and R. Nowakowski. "A Game of Cops and Robbers Played on Products of Graphs." *Discrete Mathematics*, **186**:253–268, 1998.

[NSW02] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. "Random Graph Models of Social Networks." In *Proceedings of the National Academy of Science of the United States of America*, volume 99, pp. 2566–2572, 2002.

[NW83] R. Nowakowski and P. Winkler. "Vertex to Vertex Pursuit in Graph." *Discrete Mathematics*, **43**:235–239, 1983.

[Par76] T. Parsons. *Theory and Applications of Graphs*, chapter Pursuit–evasion in a Graph, pp. 426–441. Springer, 1976.

[PF95] V. Paxon and S. Floyd. "Wide–area Traffic: The Failure of Poisson Modeling." *IEEE-ACM Transactions on Networking*, **3**(3), 1995.

[PP03] C. P. Phleeger and S. L. Phleeger. *Security in Computing.* Prentice Hall, 2003.

[Pro] Oregon RouteView Project. "Available at: www.routeviews.org/.".

[Pro04] N. Provos. "A Virtual Honeypot Framework." In *Proceedings of the 12th USENIX Security Symposium*, pp. 1–14, 2004.

[PV02] R. Pastor-Satorras and A. Vespignani. *Handbook of Graphs and Networks: From the Genome to the Internet*, chapter Epidemics and Immunization in scale–free Networks, pp. 113–132. Wiley–VCH, 2002.

[PVV01] R. Pastor-Satorras, A. Vazquez, and A. Vespignani. "Dynamical and Correlation Properties of the Internet." *Physical Review Letters*, **87**(258701), 2001.

[Qui85] A. Quilliot. "A Short Note about Pursuit Game Played on Graph with a Given Genus." *Journal of Combinatorial Theory Series B*, **38**:89–92, 1985.

[Ran99] D. A. Rand. *Advanced Ecological Theory, Principles and Applications*, chapter Correlation Equations and Pair Approximation for Spatial Ecologies, pp. 100–142. Blackwell Science, 1999.

[SGJ01] S. Staniford, G. Grim, and R. Jonkman. "Flash Worms: Thirty Seconds to Infect the Internet. Available at: www.silicondefense.com/flash/." 2001.

[SH82] J. F. Shoch and J. A. Hupp. "The Worm Programs—Early Experience with Distributed Computation." *Communications of the ACM*, **25**(3):172–180, 1982.

[SM01] W. Schleifer and M. Mannle. "Online Error Detection through Observation of Traffic Self–similarity." In *IEEE Proceedings on Communications*, volume 148, 2001.

[Sol90] A. Solomon. "Epidemiology and Computer Viruses. Available at: vx.netlux.org." 1990.

[SPW02] S. Staniford, V. Paxson, and N. Weaver. "How to Own the Internet in Your Spare Time." In *Proceedings of the 11th USENIX Security Symposium*, pp. 149 – 167, 2002.

[ST93] P. Seymour and R. Thomas. "Graph Searching and a Min–Max Theorem for Tree–width." *Journal of Combinatorial Theory Series B*, **58**:22–33, 1993.

[ST00] Y. Stamatiou and D. M. Thilikos. *Electronic Notes in Discrete Mathematics*, volume 3, chapter Monotonicity and Inert Fugitive Search Game. Elsevier Science Publishers, 2000.

[STA95] P. Spirakis, B. Tampakas, and H. Antonopoulou. "Distributed Protocols Against Mobile Eavesdroppers." In *Proceedings of the 9th International Workshop on Distributed Algorithms LNCS*, volume 972, pp. 160–167, 1995.

[Sto99] A. Stoimenow. *On Enumeration of Chord Diagrams and Asymptotics of Vassiliev Invariants*. PhD thesis, University of Berlin, 1999.

[SZ04] G. Serazzi and S. Zanero. *Performance Tools and Applications to Networked Systems*, chapter Computer Virus Propagation Models, pp. 26–50. 2004.

[TH03] L. S. Tsimring and R. Huerta. "Modeling of Contact Tracing in Social Networks." *Physica A*, **325**:33–39, 2003.

[VCI99] M. de Vivo, E. Carrasco, G. Isern, and G. de Vivo. "A Review of Port Scanning Techniques." *Operating Systems Review*, **29**(2):41–48, 1999.

[VVK98] M. de Vivo, G. de Vivo, R. Koeneke, and G. Isern. "Internet Vulnerabilities Related to TCP/IP and T/TCP, Internet Security Attacks at the Basic Level." *Operating Systems Review*, **32**(2):4–15, 1998.

[WDP03] A. Wagner, T. Dubendorfer, B. Plattner, and R. Hiestand. "Experiences with Worm Propagation Simulations." In *Proceedings of the 2003 ACM workshop on Rapid Malcode*, pp. 34–41, 2003.

[Wea02] N. Weaver. "Potential Strategies for High Speed Active Worms: A Worst Case Analysis. Available at: brass.cs.berkeley.edu." 2002.

[Wil02] M. M. Williamson. "Throttling Viruses: Restricting propagation to defeat malicious mobile code." In *Proceedings of the 18th Annual Computer Security Applications Conference*, p. 61, 2002.

[WKE00] C. Wang, J. C. Knight, and M. C. Elder. "On Computer Viral Infection and the Effect of Immunization." In *Proceedings of the 16th Annual Computer Security Applications Conference*, p. 246, 2000.

[WL03] M. M. Williamson and J. Leveille. "An Epidemiological Model of Virus Spreading and Cleanup." In *Proceedings of the Virus Bulletin Conference*, 2003.

[Woo99] M. Wooldridge. *Multiagent Systems*, chapter Intelligent Agents, pp. 27–77. The MIT Press, 1999.

[WPS03a] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. "Large Scale Malicious Code: A Research Agenda. Available at: www.cs.berkeley.edu/ nweaver." 2003.

[WPS03b] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. "A Taxonomy of Computer Worms." In *Proceedings of ACM Workshop on Rapid Malcode*, 2003.

[WS98] D. J. Watts and S. H. Strogatz. "Collective Dynamics of Small–World Networks." *Nature*, **393**:440–442, 1998.

[WVG04] J. Wu, S. Vangala, L. Gao, and K. Kwiat. "An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques." In *Proceedings of the Network and Distributed System Security Symposium*, 2004.

[WW03] Y. Wang and C. Wang. "Modeling the Effects of Timing Parameters on Virus Propagation." In *Proceedings of the ACM Workshop on Rapid Malcode*, pp. 61–66, 2003.

[WWS04] C. Wong, Chenxi Wang, Dawn Song, Stan Bielski, and Gregory R. Ganger. "Dynamic Quarantine of Internet Worms." In *Proceedings of the International Conference on Dependable Systems and Networks DSN-2004*, 2004.

[YJB02] S.-H. Yook, H. Jeong, and A. L. Barabasi. "Modeling the Internet's Large–scale Topology." In *Proceedings of the National Academy of Sciences of the United States of America*, volume 99, 2002.

[ZGT02] C. C. Zou, W. Gong, and D. Towsley. "Code Red Worm Propagation Modeling and Analysis." In *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 138–147, 2002.

[ZGT03a] C. C. Zou, W. Gong, D. Towsley, and D. Gao. "Monitoring and Early Detection for Internet Worms." In *Proceedings of the 10th ACM Conference on Computer and Communication Security*, 2003.

[ZGT03b] C. C. Zou, W. Gong, D. Towsley, and D. Gao. "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defenses." In *Proceedings of the ACM CCS Workshop on Rapid Malcode*, 2003.