5-2012

Modeling the Spread of Biologically-Inspired Internet Worms

Emma Strubell

MODELING THE SPREAD OF BIOLOGICALLY-INSPIRED INTERNET WORMS

by

Emma Strubell

A Thesis Submitted in Partial Fulfillment
of the Requirements for a Degree with Honors
(Computer Science)

The Honors College

University of Maine

May 2012

Advisory Committee:

David Hiebeler, Associate Professor of Mathematics, Chair
Sudarshan S. Chawathe, Associate Professor of Computer Science, Chair
François G. Amar, Associate Professor of Chemistry, Honors Faculty
Roy Turner, Associate Professor of Computer Science
Timothy Waring, Assistant Professor of Economics

Abstract

Infections by malicious software, such as Internet worms, spreading on computer networks can have devastating consequences, resulting in loss of information, time, and money. To better understand how these worms spread, and thus how to more effectively limit future infections, we apply the household model from epidemiology to simulate the proliferation of adaptive and non-adaptive preference-scanning worms, which take advantage of biologically-inspired strategies. From scans of the actual distribution of Web servers on the Internet, we find that vulnerable machines seem to be highly clustered in Internet Protocol version 4 (IPv4) address space, and our simulations suggest that this organization fosters the quick and comprehensive proliferation of preference-scanning Internet worms.

Acknowledgements

First and foremost, I would like to thank my parents for funding my college education as well as the life that led me here. I am also very thankful to Professor Hiebeler, who introduced me to this project, funded me, and helped me along the way, both on this project and in getting into graduate school. I am also indebted to Professor Chawathe for his help developing the masochism necessary to pursue, and be accepted into, graduate studies in computer science, and to Avner Maiberg for keeping me sane throughout past two years. Of course, I must thank everyone else on my committee for tolerating my constant scheduling issues and tendencies to work at the last minute, and Steve Cousins for facilitating that last minute work by helping me debug my simulations on the new, experimental cluster.

Additionally, the research presented in this thesis was conducted with financial support from a Research and Creative Achievement Fellowship granted by the College of Liberal Arts and Sciences, and from the National Science Foundation under Grant No. DMS-0746603 to David Hiebeler.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Malicious software such as viruses and worms spreading through computer networks is now a routine problem faced by all computer or network users and administrators. Attacks by such malware can cost billions of dollars per incident and spread around the world in a matter of hours or even minutes (D. Moore et al., 2002; D. Moore, Paxson, et al., 2003; Albanese et al., 2004). One formal definition of a *computer worm* is "an independently replicating and autonomous infection agent, capable of seeking out new host systems and infecting them via the network" (Nazario, 2004). Whereas computer viruses depend on users to proliferate, for instance by unknowingly forwarding an infected email or accidentally downloading software from a disreputable source, computer worms spread autonomously through networks, which generally allows them to propagate more quickly and ultimately cause much more damage.

In the past decade, worms began to use more advanced biologically-inspired strategies for seeking out potential hosts to invade (Chen & Ji, 2005; Avlonitis et al., 2007). Such tactics exploit inherent vulnerabilities in the structure of the Internet that enable today's worms to spread even more swiftly than their predecessors (Zou et al., 2006). In particular, a technique known as *local preference scanning* takes advantage of the tendency for machines that are nearby on the Internet to be running the same software, and thus share the same

vulnerabilities; when two machines are enough, such machines are often part of a homogeneous network maintained by the same individuals or organization. To quickly infect these pockets of vulnerable hosts, local preference scanning worms choose new targets from within the same network as their infected host machine more often than they choose a machine completely at random. This strategy for propagation typically results in persistent and widespread infections while entrenching extensive invasion within protected networks once a single host within a network has been compromised (Nazario, 2004; Pastor-Satorras & Vespignani, 2004). For example, one version of the Code Red worm infected more than 350,000 computers within 14 hours, for a time infecting more than 2,000 new hosts per minute (D. Moore & Shannon, 2001). In addition to preference scanning, modern worms often employ adaptive strategies, changing the way they spread throughout the course of an epidemic. Changes might occur randomly, or they could be based on the worm's success, or lack thereof, in a particular region of the Internet. An adaptive preference-scanning strategy might switch between a propagation algorithm that targets random IP addresses and one that targets localized addresses depending on how successful it is in spreading within its local network.

Although outbreaks of worms that use the aforementioned techniques are highly documented, little work has been done to elucidate the dynamics of their dispersal, and thus to identify effective methods for curbing or preventing such attacks and their consequences. To improve our understanding of how these worms achieve their extensive invasion, we partition the population of computers on the Internet into a hierarchy reflecting the structure imposed by the addressing system currently used by virtually all connected machines, then apply state-based epidemiological models to simulate the dynamics of susceptible, infected,

and recovered or removed populations. By thus simulating the spread of adaptive and non-adaptive preference scanning worms and running many instances of simulations that sample the whole parameter space of the model, we can visualize the effects of changing various conditions to ultimately gain a better understanding as to how these worms spread.

We know that the spatial clustering of suitable habitat strongly impacts the propagation of populations for population models on heterogeneously favorable landscapes in nature (Hiebeler, 2000, 2004). We found the same to be true of preference scanning Internet worms populating the habitat of vulnerable hosts on the Internet. Internet scan data suggest that web servers vulnerable to worm attacks are indeed highly clustered in Internet address space, which likely explains the success of preference scanning worms.

1.1 Background

1.1.1 The Structure and Nomenclature of the Internet

Every computer connected to a network, such as the Internet, has an Internet Protocol (IP) address. As street addresses identify the location of a home in our three-dimensional world, IP addresses identify the location of certain machines in IP address space. Every website corresponds to at least one IP address. In our scans and simulations, we make the simplifying assumption that the mapping between IP addresses and machines is a bijection.

Two IP addressing systems are currently in use: IP version 4 (IPv4) and IP version 6 (IPv6). We are interested in simulating worms that spread on IP version 4 (IPv4) networks, as opposed to the newer IPv6 standard, for two main reasons. First, IPv4 is the exclusive addressing system used by 93% of autonomous systems on the Internet as of July 2011 (RIPE

NCC, 2011). Additionally, IPv6 addressing will increase the size of the Internet address space by 2^{96} times to 2^{128} possible addresses, so the distribution of actual hosts in the address space, let alone those susceptible to a worm attack, would likely be much too sparse for preference-scanning to remain a viable worm dispersal strategy. Regardless, it is likely that IPv4 will remain in widespread use for quite a while, and consequently worm creators will continue to favor simple and effective scanning methods over the more elaborate and ambitious techniques that would be necessary to infect machines over IPv6 (Albanese et al., 2004; Bellovin et al., 2006; Chen & Ji, 2007).

IPv4 addresses are 32 bits long, divided into 4 byte-long segments called octets, since a byte is composed of 8 bits. In binary, 8 bits, can represent the decimal numbers 0–255, so an IPv4 address can be written as four such numbers in decimal, separated by dots. This is known as the dot-decimal notation for IP addresses. For example, 130.111.228.119 is the IP address currently mapping to `umaine.edu`. In fact, all IP addresses of the form 130.111.x.y and 141.114.x.y belong to the University of Maine to address whichever machines they please. Such blocks of addresses are referred to as subnets, and using Classless Inter-Domain Routing (CIDR) notation, those subnets would be called /16 subnets, since the addresses in each of those subnets share the same first 16 bits. More specifically, in CIDR notation the first subnet would be the 130.111.0.0/16 subnet, and the latter 141.114.0.0/16. The worms we are interested in try to spread preferentially to /8 and /16 subnets.

1.1.2 Two Preference-Scanning Worms

In Summer 2001, Code Red II, hereafter simply Code Red, was the first worm to successfully employ a preference scanning strategy for producing target addresses when

seeking out additional hosts. Although this work focuses on the specific strategy used by the second generation of the Code Red worm, subsequent destructive infections including Sasser, Blaster, and others have applied similar approaches to increase the rate and extent of their infection of the Internet. Code Red's strategy was as follows (D. Moore et al., 2002):

- With probability 3/8, it would perform short-range dispersal, generating random IP addresses in the same /16 subnet as that of the infected host;
- With probability 1/2, the worm would perform medium-range dispersal, generating random IP addresses in the same /8 subnet as that of the infected host;
- With probability 1/8, it would perform long-range dispersal, generating an IP address consisting of four randomly generated octets, avoiding the 127.0.0.0/8 and 224.0.0.0/8 subnets reserved for loopback and multicast routing, respectively.

The Sasser computer worm, which spread in April and May 2004, utilized a very similar dispersal strategy with a slightly different probability distribution. In August 2003, the Blaster worm used a complex dispersal strategy that included a mixture of several movement patterns. Since that time, worms generally use mixed strategies such as the above.

A second breakthrough in worm propagation was the adaptive preference scanning algorithm used by a variant of the Zotob worm in August 2005. Zotob began by selecting target IP addresses from the same /16 subnet as the infected host's address. The worm would then change its strategy based on the success of its attacks (Magee, 2007):

- If the first 32 attempts to infect local machines failed, then Zotob would start probing completely random IP addresses;

- If at least one of the first 32 local attempts was successful, then the worm would begin attacking random addresses after 512 local probes.

Zotob received a great deal of media coverage, both because the worm penetrated the networks of a number of well-known companies including CNN, ABC, General Electric and UPS, and because of the record turnaround time between Microsoft's report of a vulnerability and the dissemination of a worm exploiting that vulnerability (Richtel, 2005). Since Zotob, that turnaround time has only become smaller while propagation algorithms continue to improve. Today's worms are often released on the same day as a software vulnerability is exposed, making it nearly impossible to patch software before these "zero-day" threats attack (Bank, 2004). Increasing our understanding of how worms like Code Red and Zotob spread throughout the vulnerable Internet will shed light on how we can minimize the damage caused by similar worms to come.

1.1.3 State-based Epidemiological Models

When modeling epidemics, it is useful to compartmentalize the population into a number of possible states, and try to understand when and how individuals transition between those states. The two such models that we applied to the spread of worms on the Internet are the Susceptible–Infected–Recovered (SIR) and Susceptible–Infected–Recovered–Susceptible (SIRS) models, depicted as state diagrams in Figures 1.1a and 1.1b. In these models, each individual is in one of three states: Susceptible, Infected, or Recovered/Removed. Susceptible individuals become infected at a certain rate, and infected individuals either recover, in the case of the SIRS model, or are removed from the population, as in the SIR model. In the SIR model, a removed individual can never become infected again whereas a

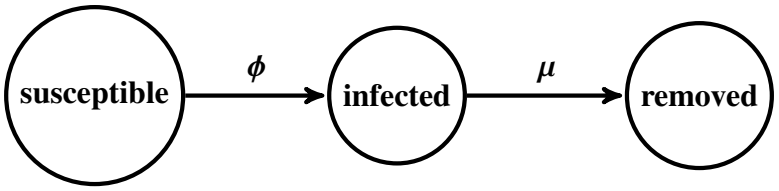
recovered individual in the SIRS model can become susceptible and eventually infected all over again.

With respect to human disease, the difference between SIR and SIRS would be akin to the difference between an outbreak of chicken pox in a school, where the population is relatively fixed with few individuals entering and leaving, and *Staphylococcus* infection in a hospital, where population turnover is high. In a school, once all the susceptible individuals have become infected and recovered, there will be no more hosts for the infection to target. In a hospital, however, a recovered individual will leave the population to be replaced by another individual who may well be susceptible.

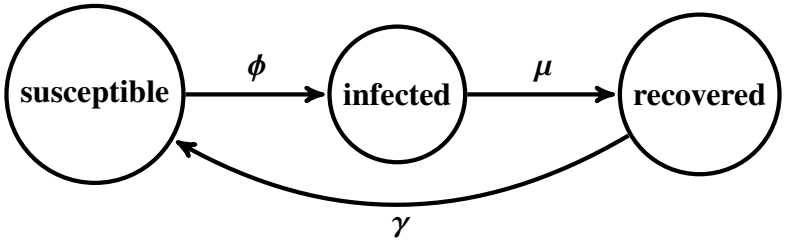
In our model of the Internet, a susceptible host is one that broadcasts itself to be running software that can be exploited by the Internet worm in question. Removal in the SIR model corresponds to a system administrator removing the worm and the vulnerability the worm used to infect the machine, so that recovered hosts may not become reinfected. SIRS recovery, on the other hand, is only temporary, assuming greater turnover of machines and software on the Internet; administrators are replacing recovered or removed machines with those running susceptible software. Because recovered and removed hosts are effectively immune, we also use this state to represent hosts that do not respond to connection attempts or whose Internet address may not correspond to a machine at all.

1.1.4 The Household Model

We use the household model in conjunction with the aforementioned state-based models to represent a hierarchical structure in the population. In the household model, the population is divided into hierarchical groups, and certain events may occur only between individuals



(a) SIR: Each infected individual attempts to infect others at rate ϕ, and is removed at rate μ.



(b) SIRS: In addition to the transitions in the SIR model, each recovered individual becomes susceptible again at rate γ.

Figure 1.1: State diagrams depicting the SIR and SIRS epidemiological models.

who are in the same group. In our household model, we divide the population into *individuals* within a *household* within a *neighborhood* within a *region*, as depicted in Figure 1.2. If we were modeling the spread of a cold between children in a school, these groups might correspond to children within families within classes within the school. In the Internet model, the hierarchical groups consist of individual machines (/32 subnets) within /16 subnets within /8 subnets within the entire IPv4 address space (the /0 subnet). These partitions arise naturally from the four-octet format of IP addresses and delineate the different spreading distances of all known preference scanning worms.

When we integrate the household model with the SIR and SIRS models, the hierarchical limitations that we enforce are three different distances of infection, as seen in Figure 1.2. Short-distance infection is between two individuals in the same household, or /16 network, medium-distance infection is between individuals in the same neighborhood, or /8 subnet,

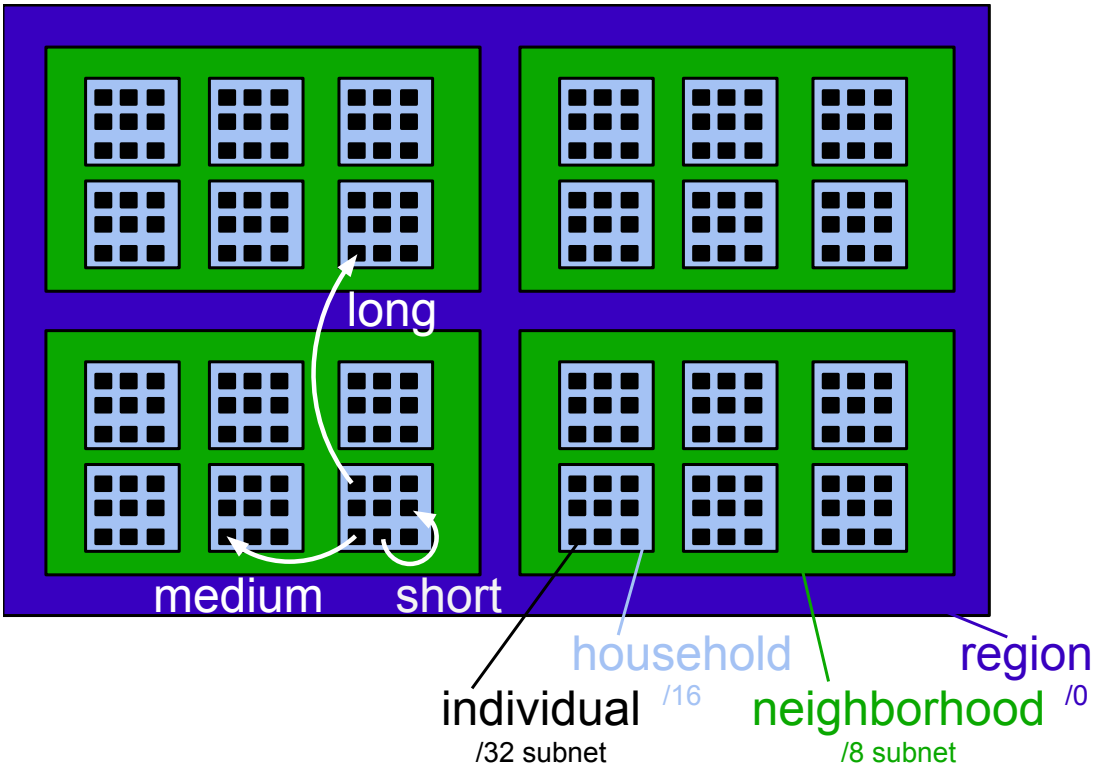


Figure 1.2: A visual depiction of the hierarchical subdivisions in our household epidemiological model, as well as the three different infection distances.

and long-distance infections occur between individuals in the same region, or /0 network, the entirety of the Internet. These dispersal distances correspond to those used by Internet worms such as Code Red and Zotob. As discussed in Section 1.1.2, Code Red would attempt to infect individuals within the same /16 subnet, short-distance infections, 3/8 of the time, /8 subnets, medium-distance infections, 1/2 of the time, and the remaining 1/8 of infection attempts would be long-distance, or anywhere in the Internet. Zotob worms would begin by only attempting short-distance infections, and switch to long-distance infections depending on the success of its initial attempts.

1.2 Related Work

Since the emergence of Internet worms in the late 1980s, many attempts have been made to characterize their proliferation. Just a few years after Morris, the first computer worm, infected thousands of university and corporate machines (Eichin & Rochlis, 1989), IBM sponsored a three year comprehensive study on the dynamics of spread and possible dangers of these unprecedented Internet infections. Among the earliest to apply models from biological epidemiology to the spread of Internet worms, Kephart et al. (1993) quickly determined that the simplifying assumption of heterogeneous contact used in many traditional epidemiological models could not be employed in the case of Internet disease. A significant factor that determines the dissemination of a worm is the topology of the network on which it is programmed to spread, which may include peer-to-peer file sharing networks, email contact lists, or the IP addressing system; just as the spatial distribution of a population changes the dynamics of an epidemic (Duryea et al., 1999), so too can the topological distribution of computer hosts on a network have a major impact on the success, or failure, of an Internet worm epidemic. Random graphs were used to simulate the small software sharing networks that served as a vector for early worm infections (Kephart & White, 1991, 1993), and the spread of epidemics on small-world social networks has been used to model the spread of email worms (C. Moore & Newman, 2000; Zou et al., 2004). The topology of certain social networks as well as the connectivity of the Internet as a whole has recently been described by many as a scale-free network. Consequently, a great deal of work has been done regarding the spread of disease, both biological and computer, on scale-free networks (Zou et al., 2004; Doyle et al., 2005; Albert & Barabási, 2002; Pastor-Satorras

& Vespignani, 2001, 2004; Newman, 2002; Barthélemy et al., 2005; Hwang et al., 2005; Balthrop et al., 2004). Others have also looked at hierarchical (Wang et al., 2000; Grabowski & Kosiński, 2004) and random (Keeling, 2005) network models with clustering, but not in conjunction with the topology of the IPv4 Internet.

A number of different types of worms have been classified based on the way that the worm identifies new victims to infect (Staniford et al., 2002; Weaver, 2002), and the effectiveness of these different strategies assessed (Zou et al., 2006; Vogt, 2004). Random scanning worms, those that attempt to infect completely random IP addresses, have been modeled a great deal due to their simplicity (D. Moore, Shannon, et al., 2003; Chen et al., 2003), while less work has been done to characterize more refined methods of dispersion, such as the sequential, hitlist, divide-and-conquer or preference scanning techniques. Chen et al. (2003) assumed homogeneous distribution of susceptible hosts throughout the Internet in their model of local preference scanning worms, and Avlonitis et al. (2007) developed a model for such worms on a /16 network made up of smaller networks of varying sizes, but with no spatial correlation. Existing models from plant epidemiology that incorporate multiple dispersal distances (Filipe & Maule, 2004; Filipe & Gibson, 1998) are difficult to apply to preference scanning worms due to the models' absence of an underlying network topology. To study the spread of preference scanning worms throughout the IPv4 address space, we employ a hierarchical household model (Barbour, 1978; Dye & Hasibeder, 1986; Daley & Gani, 1994; Kotliar & Wiens, 1990; Hiebeler et al., 2011) with clustering of susceptible individuals as measured from scans of the Internet.

Chapter 2

METHODS

2.1 Measuring the Internet

We measure the Internet by scanning batches of 5,655,600 IPv4 addresses: 100 addresses from each routable /16 subnet, with the last two octets chosen uniformly at random. We exclude certain unroutable subnets from our scans based on the special use IPv4 addresses defined in RFC 5735 (Cotton & Vegoda, 2010), all of which are listed in Table 2.1. Some unroutable subnets are /24 subnets, so they make up only a part of a /16 subnet. In this case, we still generate 100 addresses from that /16 subnet, but exclude addresses that fall within the /24 subnet. Each scanned address is unique, and we assume that each address corresponds to a single machine in our model.

When we scan a machine, we simply ask for a file commonly used by search engines to facilitate Web site indexing. Specifically, we perform an HTTP GET for the file `robots.txt` on port 80, the standard HTTP port, and record the contents of the Server response-header field, if there is one. Addresses are scanned 900 at a time, and we attempt to connect to one of Google's IP addresses (74.125.113.99) preceding each batch of 900 to ensure that our network connection is still active. In order to avoid setting off intrusion or infection detection software at organizations who might be in charge of an entire /8 or /16 network,

we take care to scan addresses in an order such that each /8 and /16 network is scanned as infrequently as possible.

We identify three types of response based on that field: *server*, *empty*, and *error*. A *server* response is one in which the Server field contains some non-whitespace text. For example, a typical response from a Microsoft IIS server might be: `Server: Microsoft-IIS/6.0`. We record the text following `Server:` and preceding a newline character. An *empty* response is one that contains a Server field, so we suspect a server is there, but there is no non-whitespace text following `Server:` in the response. In this case we record that the machine responded with an empty Server field. The final possibility is an *error* response, which lumps together all attempted connections that do not receive a response within two minutes. This includes both attempts that receive no response before the two minutes is up, which we record as a "time out," and those in which the host refuses the connection.

With this scan data, we can determine whether machines are indeed clustered in IPv4 address space. We define clustering of susceptible machines, $Q_{S|S}$, as the probability that, given a randomly selected susceptible individual in a neighborhood or a household, another randomly selected individual in the same neighborhood or household is also susceptible, a representation of percieved clustering following Hiebeler (2006). We call the probability regarding individuals in a neighborhood, $Q_{S_n|S_n}$, neighborhood-level clustering, and regarding those in a household, $Q_{S_h|S_h}$, household-level clustering. We derive the equation for neighborhood-level clustering by summing over the the probability of selecting the first susceptible individual from a neighborhood by the probability that a second individual selected in that neighborhood is also susceptible. The probability that a randomly selected individual in neighborhood j is susceptible is S_j, where S_j is the proportion of susceptible

individuals in /8 subnet j, computed as the sum of the proportions S_{ij} of the individuals in each /16 subnet within the /8 subnet i:

$$S_i = \sum_{j=0}^{255} S_{ij}$$

The probability that the first individual is selected from neighborhood j is:

$$\frac{S_j}{\sum_{i=0}^{255} S_i}$$

Multiplying this value by the probability that another individual selected from neighborhood j is susceptible, S_j, and summing over the values of all the neighborhoods, gives the overall measurement of neighborhood-level clustering:

$$Q_{S_n|S_n} = \frac{\sum_{j=0}^{255} S_j^2}{\sum_{j=0}^{255} S_j}$$

Which is equivalent to dividing the mean of the squares of the proportions of susceptible individuals in each neighborhood by the mean of the proportions of susceptible individuals in each neighborhood.

We calculate the household-level clustering similarly, except we must additionally sum over each household in each neighborhood. To measure clustering at the household level, we divide the mean of the squares of the proportions of susceptible individuals in each household by the mean of the proportions of susceptible individuals in each household. Here,

S_{ij} is the proportion of susceptible individuals in /16 subnet j, which is within /8 subnet i:

$$Q_{S_h|S_h} = \frac{\sum_{i=0}^{255} \sum_{j=0}^{255} S_{ij}^2}{\sum_{i=0}^{255} \sum_{j=0}^{255} S_{ij}}$$

We expect to find that the likelihood that two randomly selected hosts in the same /8 subnet or the same /16 subnet are both susceptible is higher than the likelihood that two hosts selected at random from the entire Internet are both susceptible. If certain types of Web servers are clustered on the Internet, then the measured values of $Q_{S|S}$ for those servers should be greater than the overall proportion of those types of servers. If $Q_{S_n|S_n}$ is greater than the overall proportion, then the Internet exhibits neighborhood-level clustering, and if $Q_{S_h|S_h}$ is greater than the overall measured proportion, then we have found household-level clustering on the Internet. Otherwise, if there is no clustering of Web servers on the Internet, then the mean of the proportions of susceptible servers would be equal to the measured values of $Q_{S|S}$.

In taking these measurements, we do not attempt to determine whether hosts are susceptible to a particular worm, and many worms propagate by exploiting vulnerabilities in software other than Web servers. Nevertheless, measuring Web servers is an unobtrusive method to obtain a sample which is likely to be representative of worm susceptibility on the Internet for a number of reasons. First, administrators running a certain type of Web server software on their networks are likely to have similar software, and thus similar vulnerabilities, running on the machines in their network. Second, since their purpose is generally serving Web pages to the public, probing Web servers for information is not intruding on its owner's

privacy without their invitation. Web servers are available via a public IP address, not hidden within a private network, and thus are available to us as well as the Internet worms we aim to simulate. Additionally, in response to routine requests, Web servers often identify their software, version, and additional associated products in the Server response-header field defined in the HTTP/1.1 standard (Fielding et al., 1999), whereas determining this information from other machines on the Internet would likely require more intrusive tactics. Although this information may be easily spoofed, we assume that the number of servers reporting incorrect or no software and version information is relatively small. Finally, even if hosts susceptible to a particular worm were randomly distributed among all hosts running a certain type of software, such machines would still be relatively clustered in the IP address space because of the clustering of hosts running that certain server software.

We simulate and thus scan the IPv4 system since it was the addressing scheme exploited by Code Red, Zotob and every Internet worm to date, and is still the exclusive addressing system used by about 93% of autonomous systems on the Internet as of July 2011 (RIPE NCC, 2011). Additionally, IPv6 addressing will increase the size of the Internet address space by 2^96 times to 2^{128} possible addresses, so the distribution of actual hosts in the address space, let alone those susceptible to a worm attack, would likely be much too sparse for even preferential scanning to remain a viable worm spreading strategy. Regardless, it is likely that IPv4 will remain in widespread use for quite a while, and consequently worm creators will continue to favor the simple and effective random scanning method over the more elaborate and ambitious techniques that would be necessary to infect machines over IPv6 (Albanese et al., 2004; Bellovin et al., 2006; Chen & Ji, 2007).

Address Block	/16 Subnets	Purpose
0.0.0.0/8	256	Broadcast messages to the current network.
10.0.00/8	256	Local communications within a private network.
127.0.0.0/8	256	Loopback addresses to the local host.
169.254.0.0/16	1	Auto-configuration between two hosts when no IP address is otherwise specified.
172.16.0.0/12	16	Local communications within a private network.
192.0.2.0/24	1/256	Assigned as TEST-NET for use solely in documentation and example source code.
192.88.99.0/24	1/256	Transmitting IPv6 packets over IPv4 networks.
192.168.0.0/16	1	Local communications within a private network.
198.18.0.0/15	2	Testing inter-network communications between two separate subnets.
198.51.100.0/24	1/256	Assigned as TEST-NET-2 for use solely in documentation and example source code.
203.0.113.0/24	1/256	Assigned as TEST-NET-3 for use solely in documentation and example source code.
224.0.0.0/4	65,536	Reserved for multicast assignments.
240.0.0.0/4	65,536	Reserved for future allocation by the IANA, except 255.255.255.255, which is reserved for local broadcasting.

Table 2.1: IPv4 addresses reserved for various purposes according to RFC 5735 (Cotton & Vegoda, 2010), adapted from the table provided by Wikipedia contributors (2012). We do not scan these addresses.

2.2 Internet Landscape Generation

An Internet landscape is a lattice, or grid, made up of 256 neighborhoods, each containing 256 households, each of which is comprised of 65,536 individuals. In the initial Internet landscape, each individual is in one of two states: susceptible or recovered/removed. A susceptible individual may become infected during the course of the simulation, whereas a recovered host is not vulnerable to the worm in question and can never be infected.

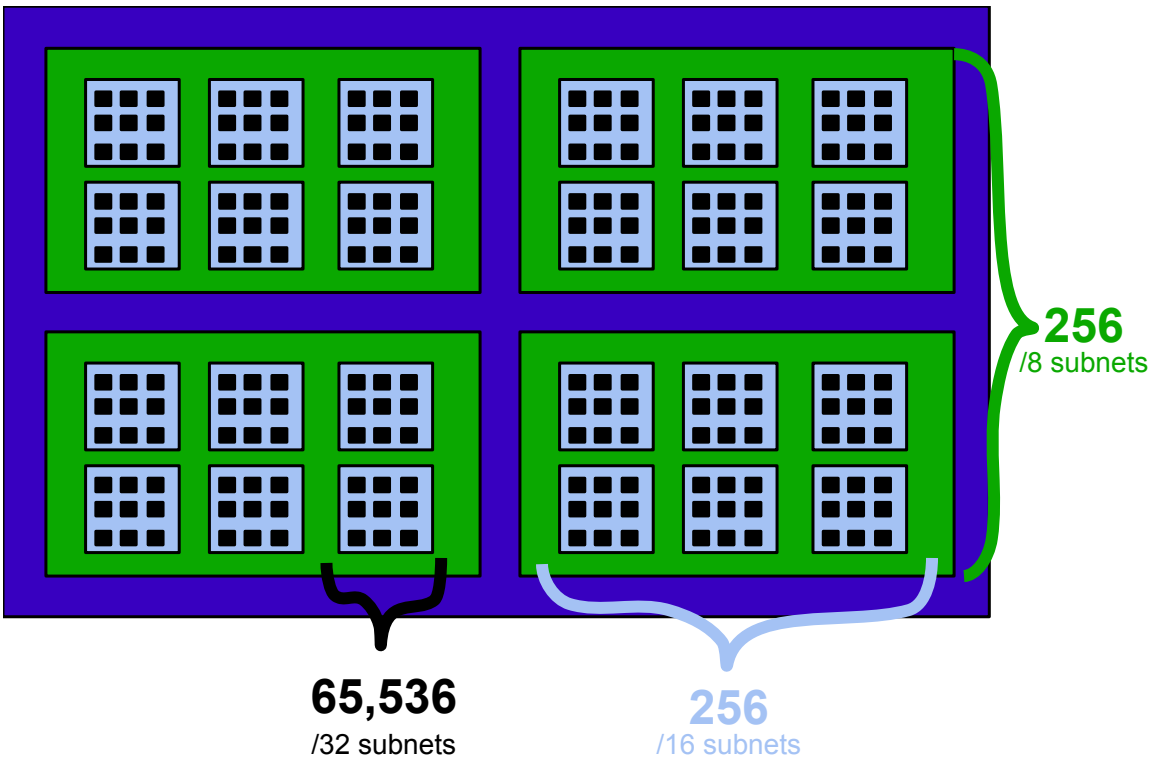


Figure 2.1: Cartoon of an Internet landscape.

We have tried two different methods for generating projected Internet landscapes based on our scan data, one method based on Bayesian statistics, and a more primitive, but slower, iterative method based on swapping locations of individuals.

2.2.1 Bayesian Method

In this method, we populate households by drawing from a bionomial probability distribution with $n = 65,536$, the total possible number of susceptible individuals per houshold, and p, the probability of the occurrence of a susceptible host, estimated from our scans of the Internet. We employ Bayesian statistics to estimate a different p for each household based on the number of susceptible individuals that we found in each /16 subnet, which we define as servers running Microsoft IIS version 6.0. In Bayesian statistics, the input parameter for a probability distribution, p in our case, is not a fixed value but rather is drawn from a probability distribution itself, which is called a prior distribution. Combining the prior distribution with the actual data, in our case the number of susceptible individuals per household, using Bayes' theorem gives the posterior distribution. The prior distribution should be chosen based on previous knowledge of the data set. However, it is important to base that choice only on concrete knowledge and not on assumptions about the data, because assuming a certain prior distribution could allow one to obtain expected results instead of actual results. For this reason, we chose a uniform prior, which assumes no prior knowledge of the distribution. Because the uniform prior distribution did not provide very accurate results, we plan to try using a Zipfian distribution as our prior distribution in the future.

Following Bayes' theorem, we can combine the prior distribution, in our case a uniform distribution, with the binomial distribution by multiplying them to compute the posterior distribution. Because the uniform distribution $U(0, 1)$ is a special case of the beta distribution $Beta(a, b)$ with $a = b = 1$, when we combine them we get a posterior distribution of $Beta(x + 1, y + 1)$, where x is the number of susceptible hosts that we found in a given

/16 network, and y the number of non-susceptible hosts in that subnet as per our Internet scans. We then use this beta distribution to generate the input parameter p for the binomial distribution which will give us an estimate of the number of susceptible individuals in each household based on our scans of the Internet.

2.2.2 Swapping Method

This method works by populating the landscape evenly with the same proportion of individuals as determined by our Internet scans, then moving those individuals around until the same level of clustering is achieved at both the neighborhood and household levels. The method is as follows.

After distributing the projected number of individuals evenly in neighborhoods and then in households, we begin by clustering individuals at the neighborhood level. To achieve this, two neighborhoods are selected at random. We compute whether moving an individual from one neighborhood to the other would increase neighborhood-level clustering, $Q_{S_n|S_n}$. If so, then we move the individual. If not, then we select two new neighborhoods and repeat the process. Individuals are swapped between neighborhoods until the same quantity of neighborhood-level clustering as measured in our Internet scans is achieved, within a small tolerance threshold. Then, we perform the same swapping at the household level, swapping individuals between randomly selected households until the household-level clustering, $Q_{S_h|S_h}$, is achieved, within the same tolerance.

Although this method is quite slow, the Internet landscapes that we produce using this method are the best we have been able to achieve, and are the landscapes that we currently use for running worm simulations. This too will be discussed in more detail in Section

3.2. We were able to improve the speed of generating these landscapes by weighting the probability of selecting a neighborhood or household to swap based on the population of individuals already in that household. This way, neighborhoods and households that have individuals to swap, and in which swaps will make more of a difference on the level of clustering, are chosen more often.

2.3 Worm Simulation

We implemented the worm simulation as a continuous-time Poisson process in the C programming language, using the OpenMPI message passing library to facilitate parallel processing of many simulations on the University of Maine supercomputing cluster. We perform many simulations in order to better understand how changing certain parameters of the simulation, such as the proportions of long- medium- and short-distance dispersal, affect the success of the worm. Table 2.2 lists the parameters that are input to the simulation.

Parameter	Description
ϕ	Rate of infection attempts
μ	Rate of recovery/removal
γ	Rate of becoming susceptible following recovery
α_0	Proportion of Type A long-distance infections
α_1	Proportion of Type A medium-distance infections
α_2	Proportion of Type A short-distance infections
β_0	Proportion of Type B long-distance infections
β_1	Proportion of Type B medium-distance infections
β_2	Proportion of Type B short-distance infections
I_0	Initial number of infected individuals
s	Probability that a failed infection attempt causes a type switch

Table 2.2: Possible input parameters for a worm simulation. If $s \neq 0$, then the simulation is adaptive. The epidemiological model to be used, SIR, SIS, or SIRS, is also specified. SIRS is the only model that uses γ.

A continuous-time Poisson process models a sequence of events happening in time. The interval of time between each event is drawn from an exponential distribution with a mean inverse to the total rate of events, and several different types of events are possible. Each type of event occurs at a certain rate, and the total rate is the sum of the rates of all the different types of events. The type of event that occurs at each time step is selected at random weighted by the rate of that event; events that have a greater rate will thus occur more frequently.

In our model, we define six possible events that can occur at each time step: long-distance infection, medium-distance infection, short-distance infection, recovery, becoming susceptible again following recovery, and an adaptive worm type switch. A long-distance infection is an infection occurring between any two individuals on the landscape; they need not be in the same household or neighborhood. This is equivalent to an Internet worm infecting a host residing at an IP address with all four octets generated randomly from a uniform distribution, the most common method of target address generation used by real Internet worms. A medium-distance infection describes an infection between two individuals within the same neighborhood, which corresponds to a worm infecting a host whose IP address shares the same first octet, or in other words resides in the same /8 subnet. Short-distance infections occur between individuals in the same household, simulating infection between two machines in the same /16 subnet.

In order to simulate adaptive worms such as Zotob, which would switch from one dispersal distance to another during the course of its infection, we divide the population into two types of worm, which we call Type A and Type B. The two types differ only in their dispersal distances. Type A worms are those infecting at the initial distance of infection, and

Type of event	Attempt rate	Success rate
Long-distance infection	$\phi\alpha_0 I_\alpha$	$r_{\alpha_0} = \frac{\phi\alpha_0 I_\alpha}{256^4} S$
Medium-distance infection	$\phi\alpha_1 I_\alpha$	$r_{\alpha_1} = \frac{\phi\alpha_1}{256^3} \sum_{j=0}^{255} I_{\alpha_j} S_j$
Short-distance infection	$\phi\alpha_2 I_\alpha$	$r_{\alpha_2} = \frac{\phi\alpha_2}{256^2} \sum_{j=0}^{255} \sum_{k=0}^{255} I_{\alpha_{jk}} S_{jk}$
Recovery	—	$r_r = I_\alpha \mu$
Susceptible	—	$r_s = R\gamma$
Adaptive type switch	—	$r_t = \frac{s\phi\alpha_2}{256^2} \sum_{j=0}^{255} \sum_{k=0}^{255} I_{\alpha_{jk}} (256^2 - S_{jk})$

Table 2.3: How we calculate the rates of attempts and successes of the six events in the worm simulation. These computations apply to Type A worms; the Type B calculations follow from these, replacing α by β, except in the case of an adaptive type switch, which only applies to Type A worms. Because we only simulate successful events, we use only the rates of successful events in the simulation. Recovery, becoming susceptible following recovery, and type switches are always successful, so they have no "attempt rate."

so all new infections are Type A. In the case of Zotob, Type A worms would perform only short-distance infections. Type A worms then switch to Type B, which attempt to infect new hosts at a second and final distribution of distances. For Zotob, Type B worms would perform only long-distance infections. The two types of worm are not limited to exclusively short- or long-distance infections, however. In the same way that Code Red's dispersal distances followed a simple probability distribution, attempting short-distance infections 3/8 of the time, medium-distance 1/2 of the time, and long-distance 1/8 of the time, a Type A worm in our simulation could proliferate using those proportions, and then switch to a Type B that spreads at any other three proportions as long as they sum to 1.

Normally, one would compute the rate of each event by multiplying the given rate

that corresponds to that event, for example, as listed in Table 2.3, the rate ϕ of infected individuals' attempts to infect others supplied as an input parameter to the simulation, by the current size of the population that the rate would affect. Returning to the example of infection, to determine the rate of a Type A infection we would multiply ϕ by the total number of Type A infected individuals, I_α, since the overall rate of infection is limited by the number of infected individuals capable of infecting others. To calculate the rates of other types of events, we similarly multiply by the number of individuals capable of performing the event. In our model, we must also integrate the dispersal distance into the calculation of infection rates. Because each worm has different probabilities of performing short-, medium- and long-distance infections, we compute separate rates for each infection distance. For example, to compute a short-distance, Type A infection, we would multiply ϕ and I_α by the probability that the infection is short-distance for a Type A worm, or α_2. The final expression for the rate of Type A short-distance dispersals, would be:

$$I_\alpha \phi \alpha_2$$

The equations for the rates of all other events for a Type A worm are listed under "Attempt rate" in Table 2.3. The equations for a Type B worm are equivalent except we substitute the variable β for α, replacing the parameters associated with Type A with those for Type B worms.

In our simulation of the Internet, however, that straightforward computation is not quite sufficient. Because our total population is so large, 256^4 or about 4.3 billion individuals, and the proportion susceptible so small, around about 0.07%, simulating unsuccessful

infections would decrease the speed of each simulation by orders of magnitude. Instead, we compute the rates of successful infections and use those in the simulation, so that we only simulate successful infection attempts. To do this, we incorporate the number of susceptible individuals, S, into the rate of infection. At different distances of infection, we incorporate the sum of the product of infected and susceptible individuals at that distance. For example, to determine the rate of successful Type A short-distance infections, $r_{\alpha 0}$, we would sum over the product of the number of infected and susceptible individuals in each neighborhood, giving the equation:

$$r_{\alpha_0} = \phi\alpha_2 \sum_{j=0}^{255} \sum_{k=0}^{255} I_{\alpha_{jk}} S_{jk}$$

The equations for the rates of all other successful events are listed in Table 2.3, under "Success rate." Though this method is slightly more computationally intensive at each event in the simulation, the immense decrease in the number of events required to complete the simulation more than makes up for that slight increase in operations. Recovery, becoming susceptible following recovery, and type switches are always successful, so there is no need to perform special calculations to avoid unsuccessful events of those types.

Although type switches are always successful, we must still incorporate the number of susceptible individuals in those calculations as well since the rate of switches from Type A to Type B is based on the rate of failed infections. In fact, the rate of type switches is equivalent to the rate of failed infections, except we incorporate an additional probability, s, which is the probability that a failed infection will result in a type switch. For example, because Zotob would switch from Type A to Type B after 32 failed infections, or after 512

infection attempts if at least one was successful, we simulate Zotob by choosing s such that s is greater than $1/512$ and less than $1/32$. We calculate the rate of failed infections similarly to the rate of successful short-distance infections, except instead of multiplying the number of Type A infected individuals in each household by the number of susceptible individuals in each household, we multiply by the quantity of non-susceptible individuals in each household, or $256^2 - S$. The resulting equation for the rate of type switches, r_t, is as follows:

$$r_t = \frac{s\phi\alpha_2}{256^2} \sum_{j=0}^{255} \sum_{k=0}^{255} I_{\alpha_{jk}}(256^2 - S_{jk})$$

The simulated worm infection begins in a neighborhood and household selected at random, with neighborhoods and households weighted by the number of susceptible individuals therein. This is the same procedure by which we select neighborhoods and households for infection throughout the simulation. Infections are the only type of event that occurs until the worm achieves an initial population of infected individuals, I_0, an input parameter of the simulation. We believe that the initiator of a real worm infection would similarly ensure that his or her pestilence took hold of a certain number of machines before leaving it to spread unattended, perhaps by uploading the worm to a few machines known to be susceptible, or releasing the worm into the wild via many machines at once. Once the level of infection reaches that initial threshold I_0, events occur based on their rates, which are calculated anew from the populations of susceptible, infected, and recovered or removed individuals following each event. The simulation ends when there are no more infected individuals, which is always the case in simulations based on the SIR model, or when the

number of infected individuals is relatively constant.

We assess the success of simulated worms using three measures: initial growth rate, peak, and final proportion of susceptible individuals who become infected. Measurement of the latter two values is straightforward; during the course of the simulation, we record the peak and final proportions of infected individuals. Computing the initial growth rate, however, is slightly more difficult.

We define the initial growth rate as the slope of the initial, exponential growth of the population of infected machines plotted over time. We do not include the initial, slow, often unstable increase in the infection level that sometimes occurs before the infection starts to spread rapidly in our measurement, unless the infection dies out without achieving a great spike in infection levels. Instead, we try to isolate the first great spike in infection. The difficulty thus arises in determining where to begin and end our measurement of the initial growth rate. Once we have those bounds, taking the logarithm of the data and performing a linear regression to approximate the rate of growth is straightforward. To find the beginning of the rapid initial growth period, we compare the slope of the log-transformed last ten data points to a very large threshold slope that we believe corresponds to very rapid growth. If the slope is greater than the threshold, then we continue keeping track of the simulation time and infection level, and start checking whether the slope has decreased. As soon as the current computed slope is less than the last slope that was calculated, we stop recording initial growth rate data. We do not compute the slope at every event, but rather every few events, so very small fluctuations in the number of infected individuals will not prematurely halt our recording of initial growth rate data. At the end of the simulation, we perform a final linear regression on the initial growth rate data and record that result along with the

peak and final infection levels.

Chapter 3

RESULTS

3.1 Clustering of Servers on the Internet

So far, we have scanned 45,244,800 IP addresses, slightly more than 1% of the Internet. Because the addresses we have scanned are randomly sampled from throughout IPv4 address space, we believe that our data so far provide a representative sample of the distribution of Web servers on the Internet. Our data suggest not only that Web servers are highly clustered in IP address space at both the neighborhood- and household-levels, but that servers running certain software, such as Microsoft's Internet Information Services (IIS) Web server, and certain versions of software, such as IIS version 6.0, are also clustered. Table 3.1 lists the clustering that we measured at the neighborhood and household levels of all Web servers, IIS servers of any version, and servers running IIS versions 6.0 and 7.5, the most common and newest stable versions of IIS, respectively. As listed in Table 3.1, the clustering parameter, or

Measurement	All Servers	All IIS	IIS 6.0	IIS 7.5
Household-level (/16) clustering	0.179112	0.052788	0.049187	0.036062
Neighborhood-level (/8) clustering	0.019823	0.003706	0.002399	0.000844
Proportion of total sample	0.006596	0.001067	0.000631	0.000214

Table 3.1

probability that a randomly selected machine running a certain type of software in a /8 or /16 subnet would be running the same type of software as another randomly selected individual,

was higher than the proportion of that type of software throughout the total sample in every case. Additionally, we observed clustering in every case at both the neighborhood and household levels, though the household-level clustering was always greater than that at the neighborhood-level. In fact, these trends were true for all detected versions of IIS, though we include only the most popular versions in Table 3.1.

We can also depict the computed clustering levels listed in Table 3.1 in a more intuitive map of the Internet. The clustering of Web servers of all software types and versions is depicted in Figure 3.3. We use a Hilbert curve, a space-filling fractal curve illustrated in Figure 3.1, to map the first and second octets of IP addresses to a grid while maintaining spatial correlations.

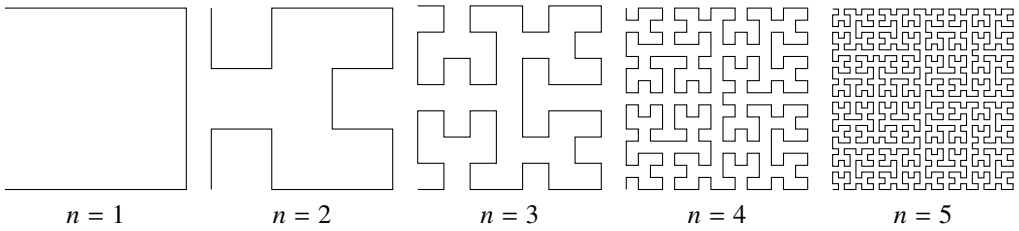


Figure 3.1: Hilbert curves of orders 1–5. In our depiction of the Internet, we draw each /8 subnet as an order 4 Hilbert curve composed of its /16 subnets, and the entire Internet as a Hilbert curve of order 4 composed of those /8 subnets.

By using a Hilbert curve to map IP addresses, /16 subnets with consecutive second octets, such as the /16 subnets 130.110.0.0/16, 130.111.0.0/16, and 130.112.0.0/16, will appear next to each other in a single blob. Similarly, /8 subnets with consecutive octets, such as 129.0.0.0/8, 130.0.0.0/8, and 131.0.0.0/8, will also remain grouped on the map, as depicted in Figure 3.2.

On the Internet map, each small dot on the grid represents a /16 network, and each larger square, such as those seen in gray, is a /8 subnet. We map IP addresses to a space-filling

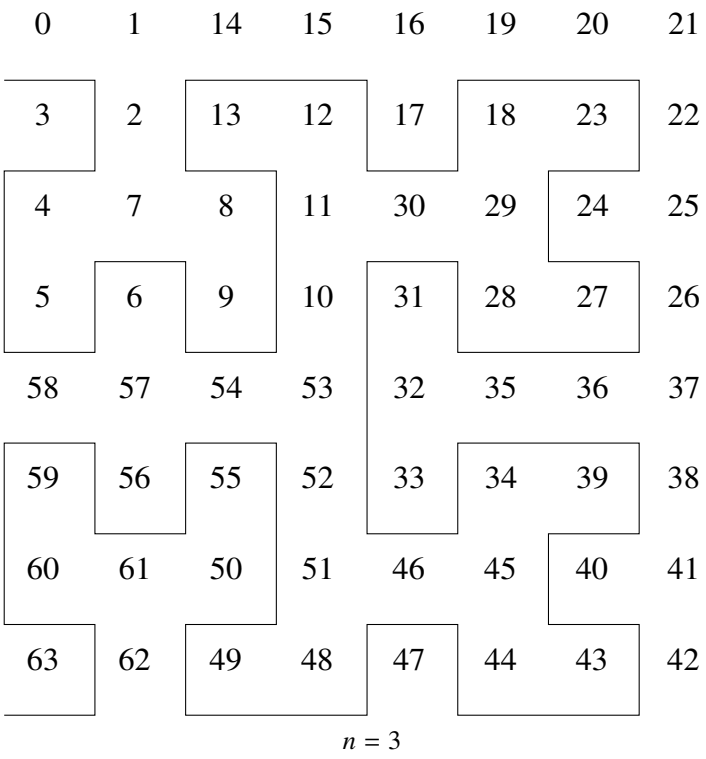


Figure 3.2: The labels on this order 3 Hilbert curve correspond to the Hilbert distances at each point on the curve, or the distance from the beginning. We visualize the Internet by interpreting the first and second octets of the scanned /16 subnets as Hilbert distances.

Hilbert curve to preserve spatial correlations. Colors represent the percent of scanned addresses that we identified to be servers, which we plot on a logarithmic scale due to the extreme differences between proportions of servers in different households; the hotter the color, the more servers, with the darkest red meaning that 99% of the addresses that we have scanned so far in that household are servers. The gray areas correspond to addresses that we do not scan because they are reserved for various reasons, as listed in Table 2.1. Single red or orange dots represent single /16 subnets that are dense with Web servers compared to their neighbors, or household-level clustering. Larger blobs of hotter colors represent clusters of /16 subnets, most often within /8 subnets, containing many servers. These blobs, because they are often smaller than an entire /16 subnet, represent some level of clustering

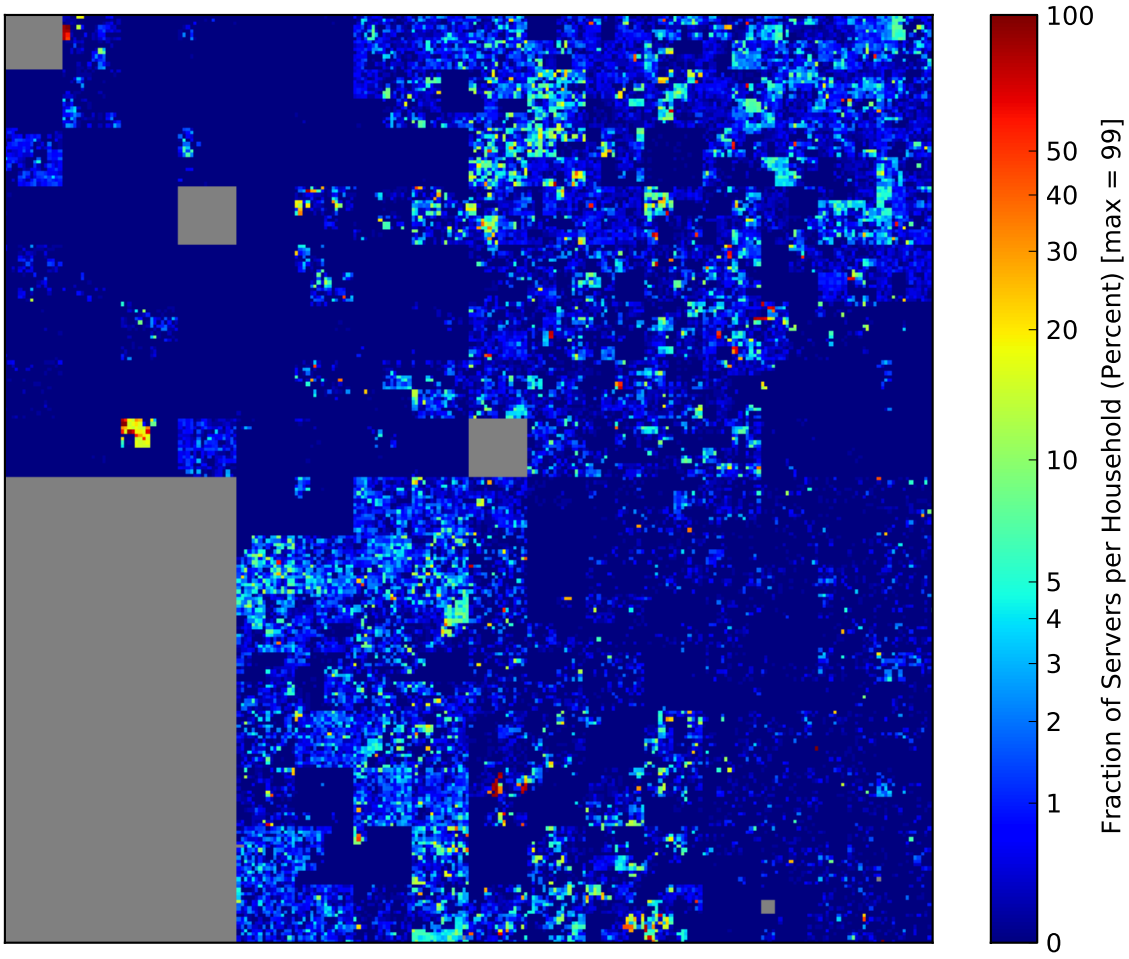


Figure 3.3: Our current map of Web servers of all software types and versions from our scans of the Internet. Refer to text for details.

between the household and neighborhood levels. Clustering at the neighborhood level is certainly present, but less stark than clustering at the sub-neighborhood and household levels; there are no entire neighborhoods containing very high density households, though many neighborhoods are filled with low-density households, where we have detected that about 5% of IP addresses correspond to servers. At the smaller spatial scales, clustering of

households containing 20–99% Web servers is more common.

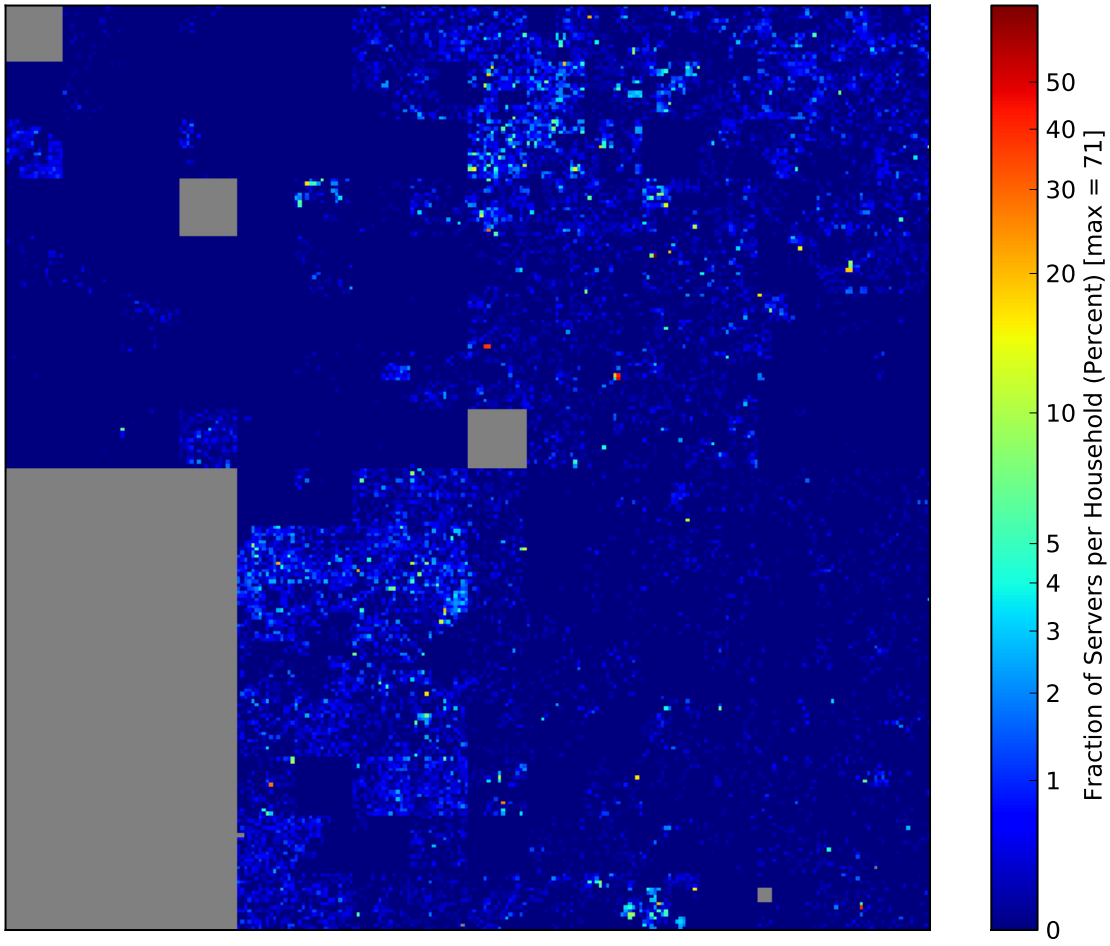


Figure 3.4: Our current map of Microsoft's Internet: Clustered IIS servers.

IIS servers show clustering similar to the overall patterns of Web servers, although there are less of them; the most populous household contains only 71% IIS servers. Still, as listed in Table 3.1, we see clustering at both the neighborhood and household levels, but much more at the household level than in neighborhoods.

3.2 Simulating the Structure of the Internet

As described in Section 2.2, we experimented with two different methods of generating Internet landscapes: One method based on Bayesian statistics, and an iterative method based on swapping individuals between neighborhoods and households. The iterative method was the first method that we tried, but that process is slow and does not produce extremely accurate simulated distributions of servers on the Internet. Although the Bayesian method is, as we expected, much faster, the landscapes that it produces are too dissimilar to the actual Internet for us to use them as the basis for our simulations.

Consider the following two landscapes, one measured from our scans of the Internet, and one generated using the iterative method.

Figure 3.5 depicts an Internet landscape generated using the iterative method, whereas Figure 3.6 illustrates the actual distribution of IIS version 6.0 servers that we have measured. The two exhibit the same measured clustering at both the neighborhood and the household levels within 0.00001. At first glance, the two landscapes seem reasonably similar. However, note that the peak percentage of servers occupying a household in the generated landscape is 28 whereas that of the actual landscape is 71%. We have not been able to find a superior method of generating Internet landscapes to this one.

The iteratively generated landscape of Figure 3.5 is much more accurate than a landscape generated using the Bayesian method, as illustrated in Figure 3.7. The nice property of this landscape is that it properly encapsulates an additional level of spatial distribution as measured from the Internet, since the susceptible population of each household is generated based on the measured population of susceptible individuals from that household. This

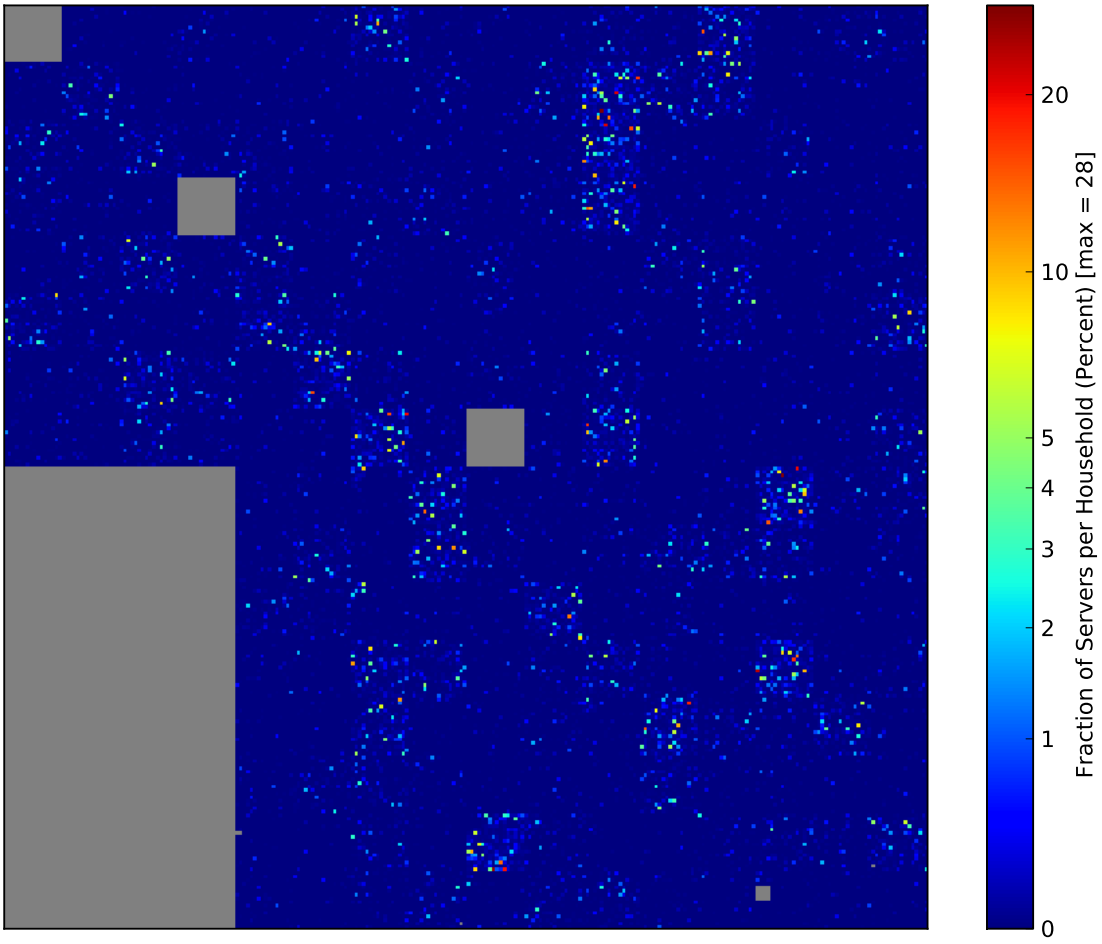


Figure 3.5: An Internet landscape generated using the iterative method, based on our detection of IIS version 6.0 servers. This is the type of landscape on which we perform simulations.

method also generates landscapes very quickly, limited only by the efficiency of our beta and binomial random number generators. Clearly, however, this method has a serious problem, which is based on our incorrect assumption of a uniform prior distribution. For that reason, the more populated a neighborhood as determined by our Internet scans, the more uniform the distribution of susceptible hosts in that neighborhood, which simply does

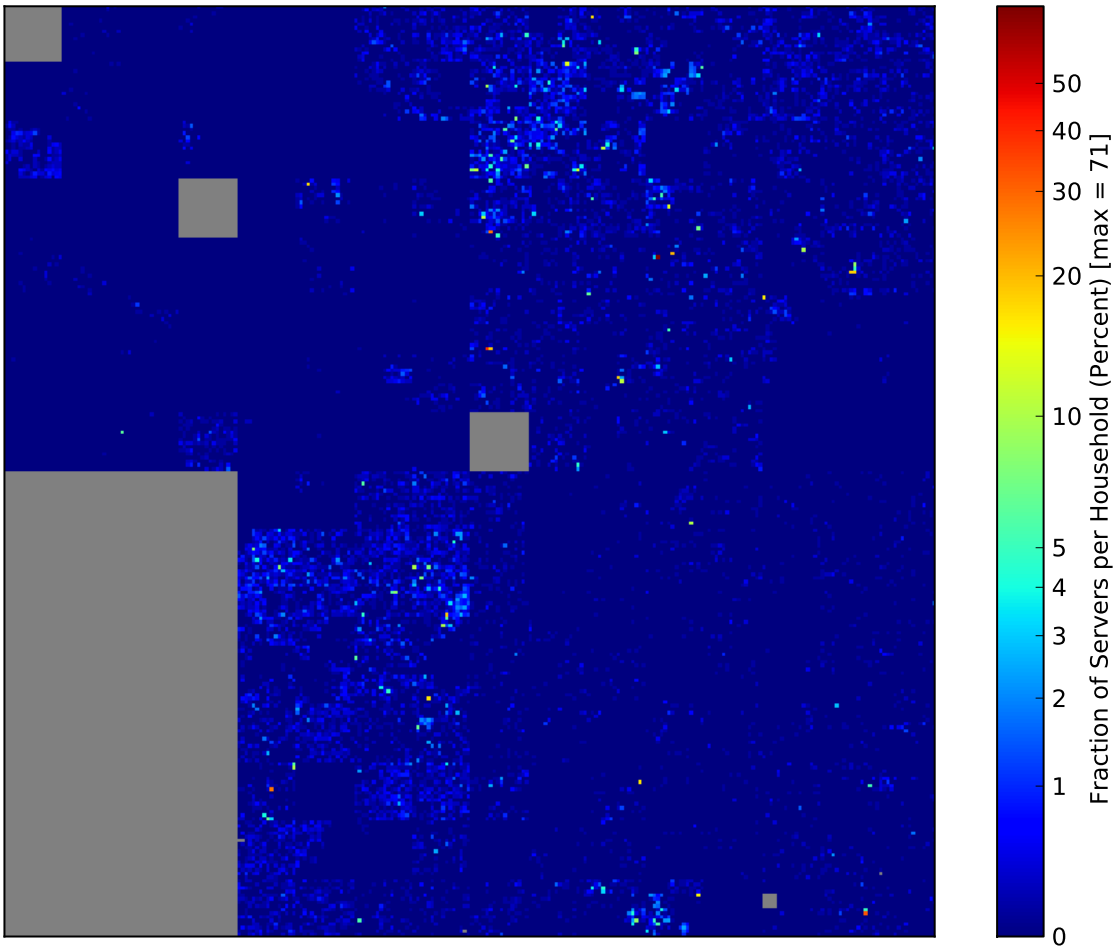


Figure 3.6: An Internet landscape depicting all the servers we detected running IIS version 6.0

not model the distribution of hosts on the Internet. We suspect that replacing the uniform prior distribution with something more similar to the power-law distribution of hosts that we see from our scans, such as a Zipfian distribution, would significantly improve the accuracy of the Bayesian generated landscapes. Until we make those changes, we will continue to use the iterative swapping method for generating Internet landscapes.

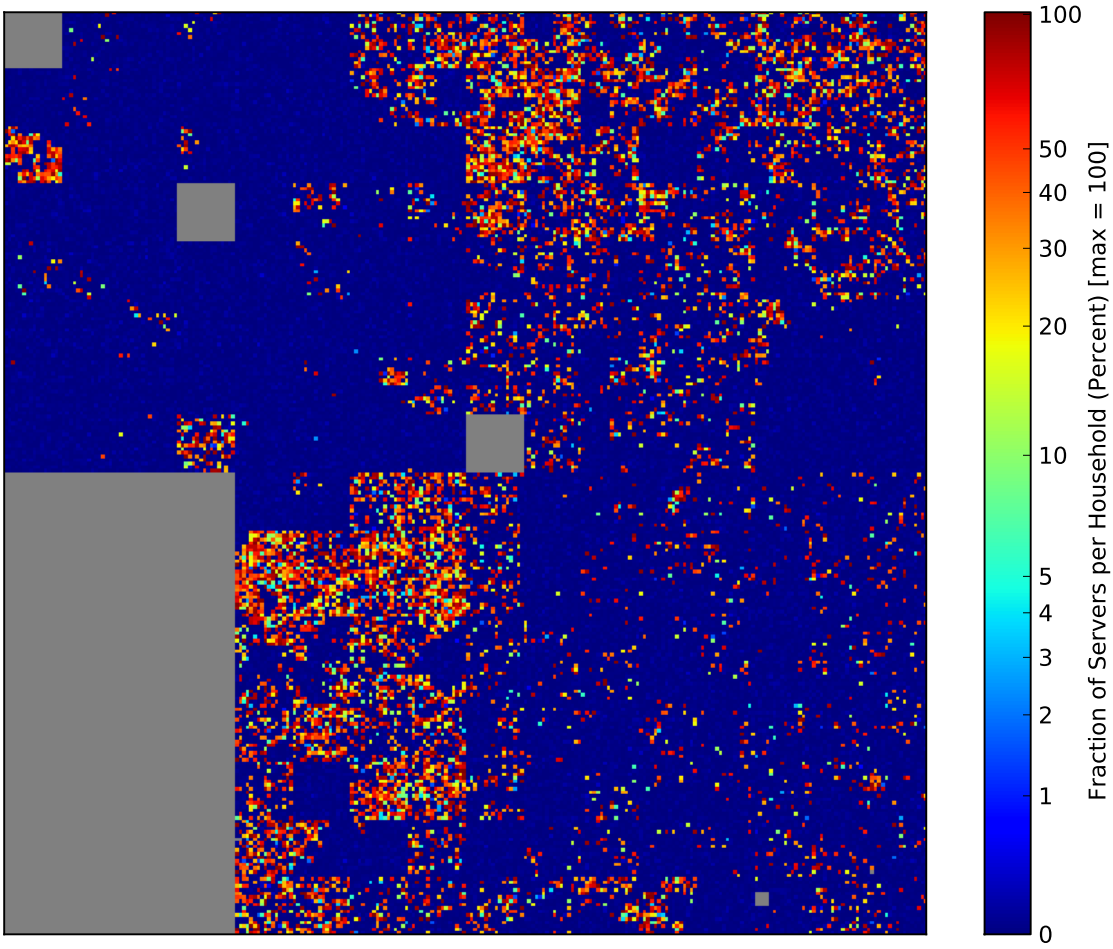


Figure 3.7: An Internet landscape generated using the Bayesian method.

3.3 Worm Proliferation

Our simulations of both adaptive and non-adaptive worms suggest that local preference-scanning is an effective strategy, but that mixing other levels of dispersal is important for the robustness of the worm. Figure 3.8 depicts the success, measured in terms of initial growth rate and peak proportion of susceptible individuals infected, of 1040 worms simulated using

the SIRS model. Each dot on the plot represents a single worm simulation. Except for Code Red and Zotob, whose data points are marked in black and gray, respectively, each worm's probabilities of short-, medium-, and long-distance infections were generated at random. We color each data point to represent the distribution of short-, medium-, and long-distance dispersals; the more short-distance infections that a worm performs, the more red its dot, the more medium-distance dispersal, the more green, and the more long-distance dispersal, the more blue.

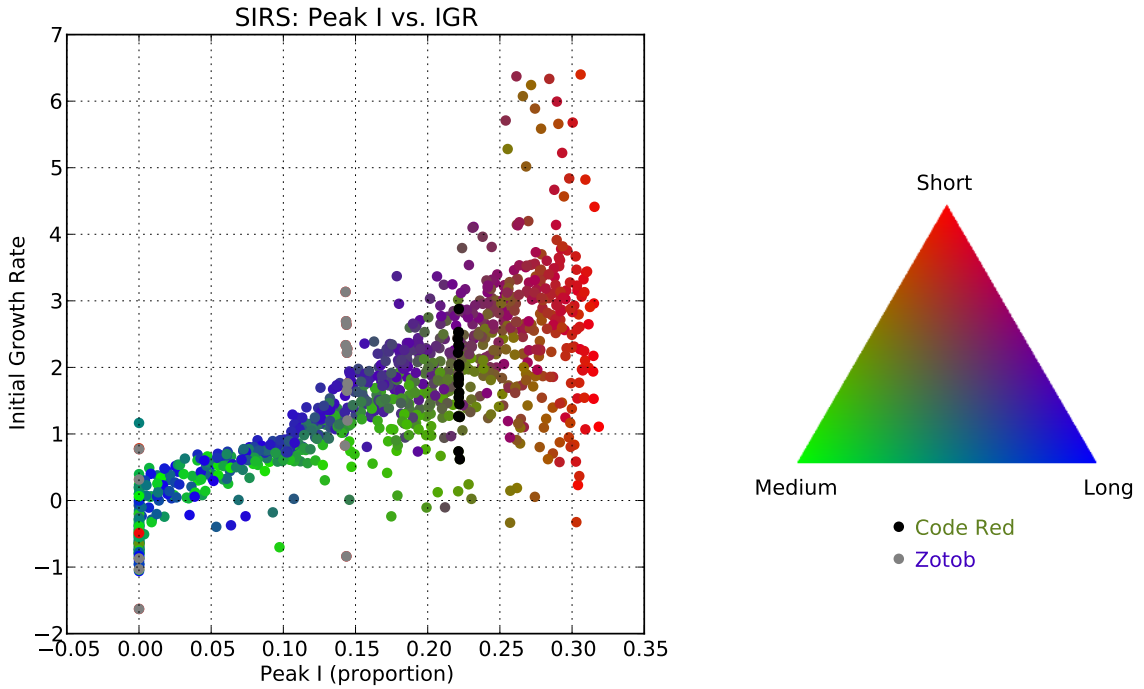


Figure 3.8: Results from 1040 SIRS simulations, plotted as peak proportion of susceptible individuals infected versus initial growth rate (IGR), plus 20 simulations of Code Red (black) and 20 simulations of Zotob with $s = 1/32$ (gray). See text for more explanation.

Our simulations strongly suggest that more short-distance dispersal can lead to both faster proliferation, as measured by the initial growth rate, and more extensive proliferation, as measured by the peak proportion of individuals infected, than worms that use more

long- or medium-distance dispersal. The trade-off, however, seems to be the robustness of the worm. Although worms performing almost all short-distance infections *can* be very successful if they happen to enter a highly susceptible pocket of the Internet, they must find that pocket in order to succeed, and that can take longer when the worm performs less long- and medium- distance dispersal that infects other areas of the Internet. This explains the huge variation in initial growth rates of worms performing mostly short-distance dispersal. Even worms that end up infecting a large proportion of susceptible hosts may take a long time to find enough highly susceptible areas to quickly infect, and some near-100% short-distance worms fail to find such a pocket completely, perhaps beginning in a highly unsusceptible area and dying out quickly.

Indeed, looking at the proportion of long- or medium-distance infection used with short-distance infection versus the resulting initial growth rate, as depicted in Figure 3.9, short-distance dispersal has the greatest impact on initial growth rate, but the most successful worms use a combination of short-, medium-, and long-distance infections. Although short-distance infections seem to be important for achieving a fast initial growth rate, the worms with the highest initial growth rates spread using a combination of short-, long- and medium-distance probes. Moreover, worms using the highest proportions of short-distance dispersal often perform no better than those performing only medium- or long-distance infections, as seen in the bright red points interspersed with bright blue points on the plot of medium-distance dispersal, and with the bright green dots in the plot of long-distance dispersal in Figure 3.9.

We expected medium-distance dispersal to play a more significant role in successful worm infestations, but it does not seem to provide much of an advantage over long-distance

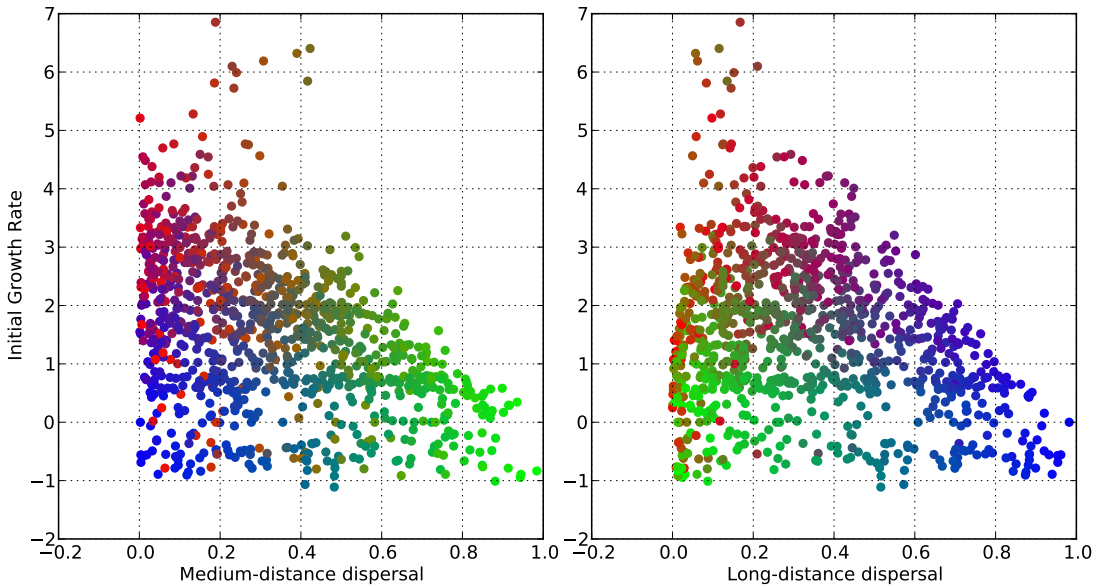


Figure 3.9: Long- and medium-distance dispersal of 1024 worms simulated using the SIRS model versus the resulting initial growth rates.

infections. Worms performing mostly medium- or long-distance infections seem to spread nearly equally well, but those performing mostly short-distance dispersal with medium-distance mixed in do seem to perform a bit better than those with long-distance mixed with the short, as is suggested by the high initial growth rates persisting further along the x-axis when looking at medium-distance dispersal as opposed to long-distance. The unexpectedly small impact of medium-distance dispersal could be explained by the smaller level of clustering that we measured, and thus simulated, at the neighborhood level compared to the household level; the two measurements persistently differ by about an order of magnitude.

We found similar trends to be true regarding the peak proportion of individuals infected in SIRS simulations. Figure 3.10 shows the amount of medium- and long-distance dispersal compared to the peak level of infection achieved by the worms in our simulations. As in the case of initial growth rate, short-distance dispersal is important for achieving a high peak

level of infection. Unlike initial growth rate, however, the worms that infected the greatest amount of susceptible hosts on the landscape were those employing the most short-distance dispersal. So, infecting as many susceptible hosts as possible relies less on a combination of short-medium- and long-distance dispersal, and more on short-distance dispersal quickly infecting as many hosts as possible in susceptible patches of the Internet.

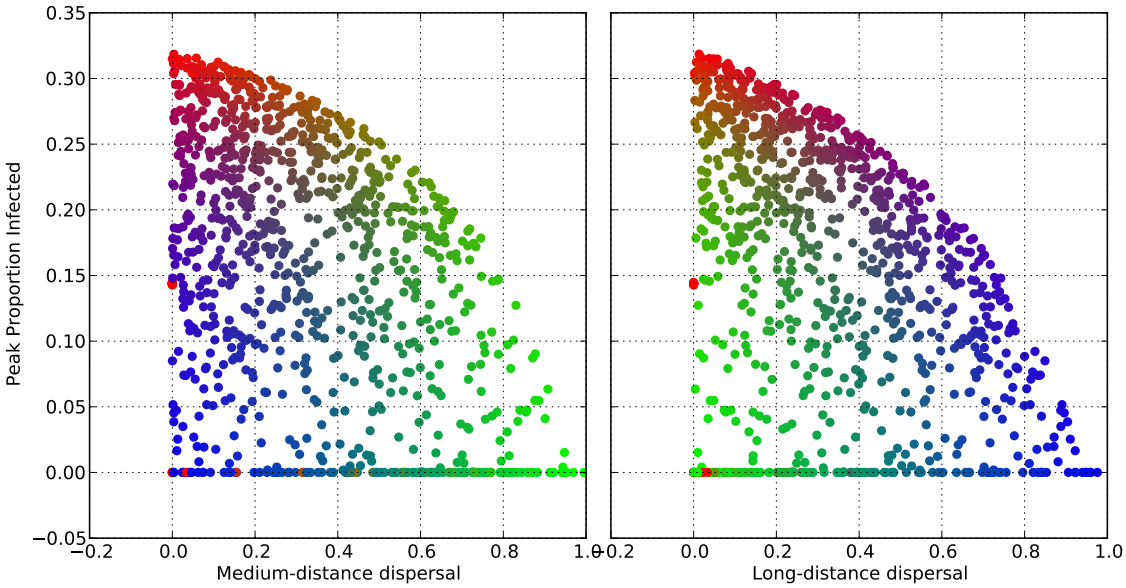


Figure 3.10: Long- and medium-distance dispersal of 1024 worms simulated using the SIRS model versus the resulting final proportion of infected individuals.

Regarding the success of Zotob versus Code Red in the SIRS model, although Zotob achieved a similar range of initial growth rates as Code Red, its peak level of infection was consistently lower than that of Code Red. This likely follows from the above analysis. In the Zotob simulations pictured in Figure 3.8, the probability that a given infection will cause a type switch, or s, was $1/32$. Because Zotob would switch from short- to long-distance dispersal after 32 short-distance attempts if all of those attempts failed, and after 512 attempts if at least one was successful, $s = 1/32$ is a worst-case estimate, essentially

assuming that all short-distance infection attempts are failures, which is most often not the case. When $s = 1/32$, Zotob's adaptive mixing of short- and long-distance dispersal allows it to achieve initial growth rates akin to those of Code Red, but because Zotob infections quickly switch from all-short-distance to all-long-distance dispersal distances, the majority of Zotob infections are long-distance, not short-distance, and so Zotob does not succeed in infecting as many individuals as Code Red. This distribution of long- and short-distance dispersing Zotob worms is depicted in Figure 3.11, a plot of the course of a Zotob infection over time. In SIRS Zotob infections with $s = 1/32$, short-distance dispersal only dominates

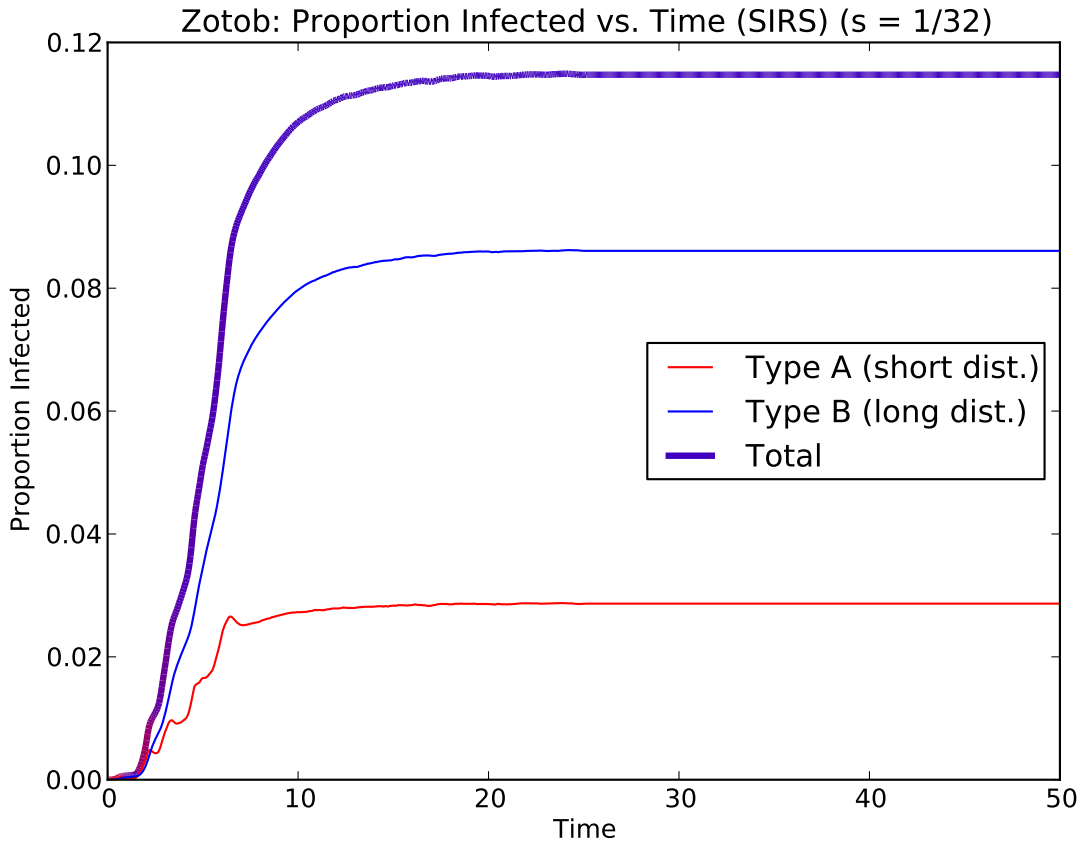


Figure 3.11: Plot of a Zotob infection versus time with $s = 1/32$. The thick line is the sum of the Type A and Type B infections, and is colored according to the proportions of Type A (red, short-distance) and Type B (blue, long-distance) worms making up the total infection. Results averaged over 10 Zotob simulations.

in the very beginning of the epidemic, long-distance dispersal worms soon taking over, with about twice as many long-distance spreading worms as short-distance at the final, stable state of the worm infection.

If we simulate Zotob with $s = 1/512$, a best-case scenario in which every series of short-distance infections results in at least one success, the results change drastically. As pictured in Figure 3.12, with $s = 1/512$, significantly more worms perform short-distance dispersal throughout the course of the simulation, which makes sense since the worms are 16 times less likely to switch from short-distance Type A to long-distance Type B. The

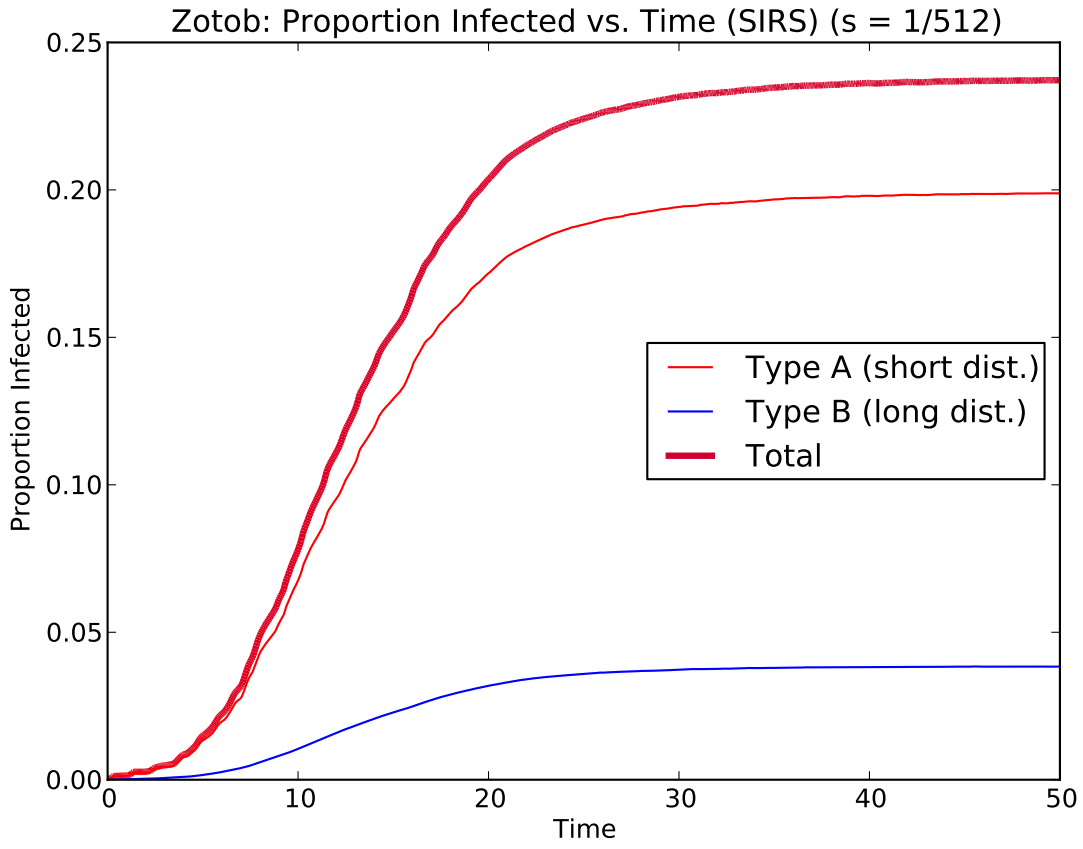


Figure 3.12: Plot of a Zotob infection versus time with $s = 1/512$. The thick line is the sum of the Type A and Type B infections, and is colored according to the proportions of Type A (red, short-distance) and Type B (blue, long-distance) worms making up the total infection. Results averaged over 10 Zotob simulations.

result of so much more short distance dispersal, still combined with a small amount of long-distance dispersal, is a worm that exceeds Code Red's peak infection level at the cost of initial growth rate. Figure 3.13 depicts the proliferation of a Code Red worm compared to a Zotob infection with $s = 1/32$ and another with $s = 1/512$. Although the Zotob

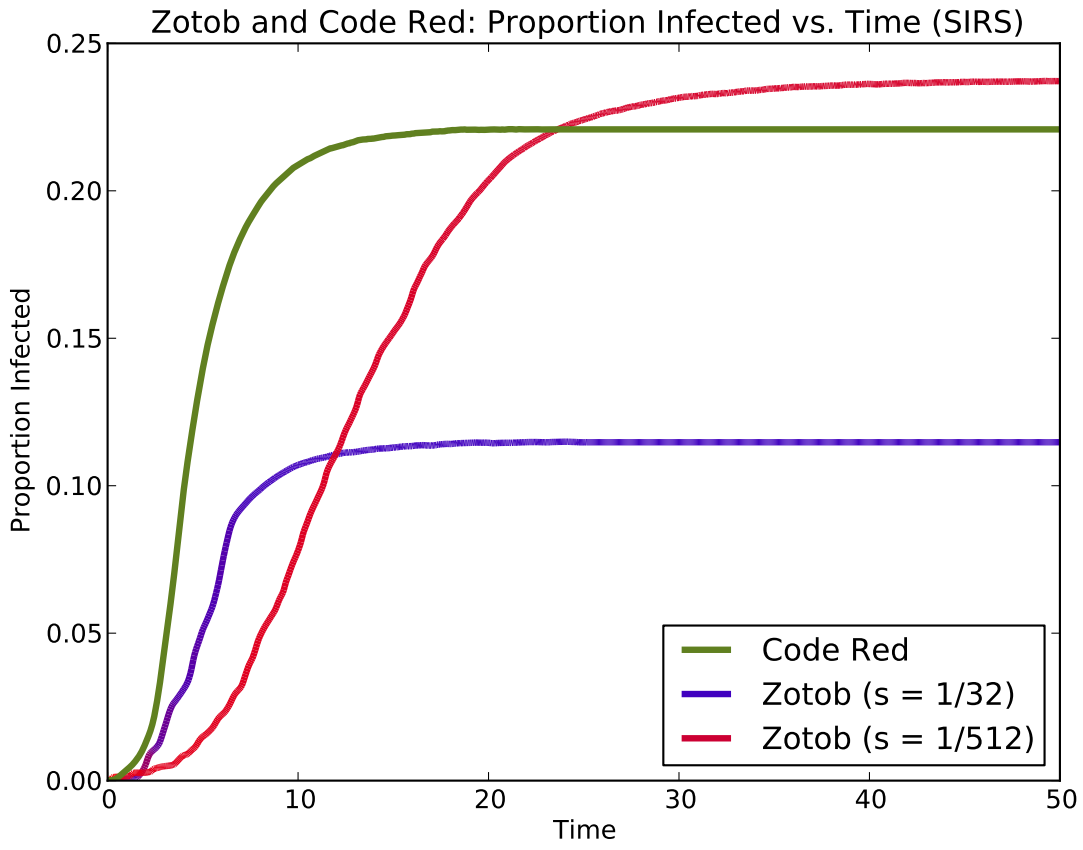


Figure 3.13: Plot of Code Red and two types of Zotob infections versus time. Results averaged over 10 simulations.

infection with more short-distance dispersal suffers from the slowest initial growth rate, it boasts the greatest overall peak infection level, beating Code Red. The two simulated extremes of a Zotob infection demonstrate the flexibility of adaptive preference-scanning worms. Although our model sets s at a fixed value throughout the simulation, in a real Zotob infection, each instance of the worm would adapt to its level of success, employing

long-distance dispersal to spread to more areas if it is not successful in its current subnet, or using more short-distance dispersal if its subnet is highly susceptible. An actual adaptive worm epidemic would thus likely achieve a near-ideal balance of growth rate and depth, since each worm modifies its strategy to best suit its area of the Internet.

Also of note are the sequential spikes visible in the initial growth period of the Type A Zotob worms, seen most distinctly in Figure 3.11. These spikes in the population of short-distance worms most likely correspond to long-distance dispersal into highly susceptible pockets of hosts on the Internet, and the resulting bursts of short-distance infections radiating through those areas. This finding strengthens our conclusion that the success of local preference-scanning worms depends on the clustering of susceptible hosts on the Internet.

Short-distance dispersal also correlated with success in our simulations using the SIR epidemiological model. Figure 3.14 depicts the results of 1040 SIR simulations using the same visualization scheme as described for Figure 3.8. The relative success of worms using different amounts of short-, medium- and long-distance dispersal in SIR epidemics is very similar to that of worm epidemics modeled using the SIRS model. The main difference between the two is that the success of worms modeled using SIR seems more erratic; Code Red and Zotob's successes, for example, plotted in black and gray respectively in Figure 3.14, varied much more than in the SIRS simulations in terms of both peak proportion of infected individuals and the initial growth rate. This makes sense based on the differences between the two models. In SIR, recovered individuals cannot become susceptible again. As a result, when a worm spreads within a highly susceptible pocket of the Internet, the infections within that pocket will eventually die out once all the susceptible individuals have become infected. The worm will not be able to remain viable in that area, but must spread to a

different susceptible area in order to stay alive, continue increasing the proportion of infected individuals, and spread at a fast rate. Thus the success of a worm in the SIR model is much more dependent on the random number generator determining which neighborhoods and households to which the worm will spread. In SIRS simulations, on the other hand, because recovered individuals can become infected again, worms essentially have a continuous supply of susceptible hosts to infect, which gives them more time to linger in a susceptible area, and thus more chances to happen upon another highly susceptible area. The result is that worms in the SIRS model are more likely to find the available susceptible pockets in the internet to infiltrate, to do so before dying out, and ultimately to enjoy a more stable success in their infection of the Internet.

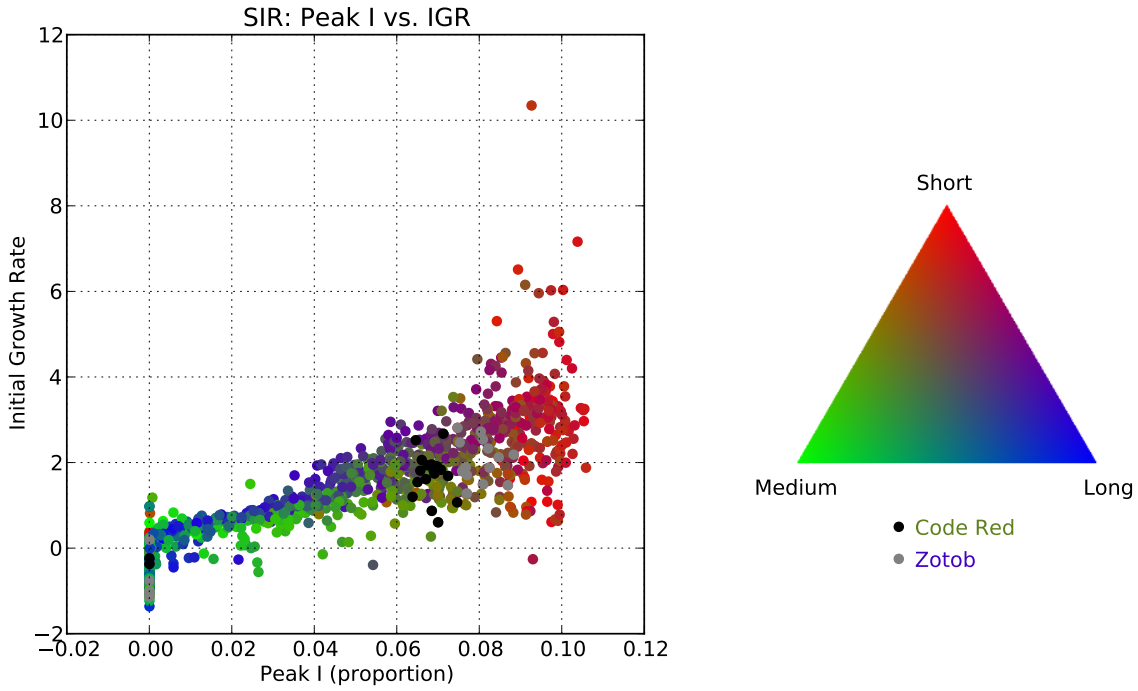


Figure 3.14: Results from 1000 SIR simulations, plotted as peak proportion of susceptible hosts infected versus initial growth rate (IGR), plus 20 simulations each of Code Red (black) and Zotob with $s = 1/32$ (gray).

Another notable difference between the SIR and SIRS simulations is the success of Zotob with $s = 1/32$ compared to Code Red. Whereas Code Red prevailed in terms of peak infection in the SIRS simulations, Zotob took the lead in SIR. In behavior exactly the opposite of SIRS, SIR simulations of Zotob with $s = 1/512$ performed poorly compared to Code Red using the SIR model. Figure 3.15 depicts the course of an SIR simulation of a Zotob worm with $s = 1/512$ using the same coloring scheme as in the SIRS figures. We

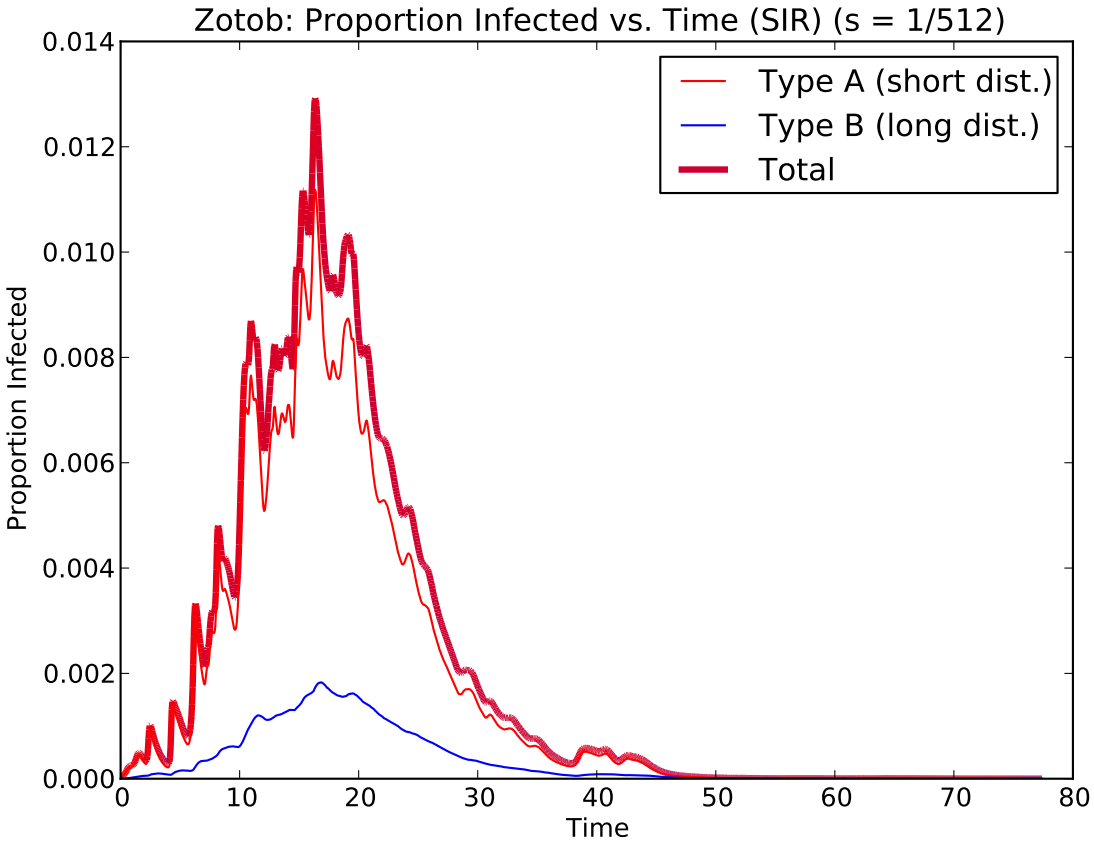


Figure 3.15: Plot of a Zotob infection versus time with $s = 1/512$. The thick line is the sum of the Type A and Type B infections, and is colored according to the proportions of Type A (red, short-distance) and Type B (blue, long-distance) worms making up the total infection. Results averaged over 10 Zotob simulations.

expect that this is due to the same reasons as the more erratic overall success discussed above. Although Zotob infections with $s = 1/512$ did very well in the more forgiving SIRS

model, where there is more time in worm-friendly households to find a new susceptible area to infect, $s = 1/512$ results in far too much short-distance dispersal to succeed in the SIR model. The sharp spikes in the population of infected individuals throughout the simulation likely represent the proliferation, and subsequent recovery, of the worm within susceptible pockets of the Internet. Unlike SIRS, in which recovered worms can become susceptible again, worms have a much harder time maintaining a stable population in the SIR model, as is certainly true of Zotob with $s = 1/512$ depicted in Figure 3.15. Unless the worms chance

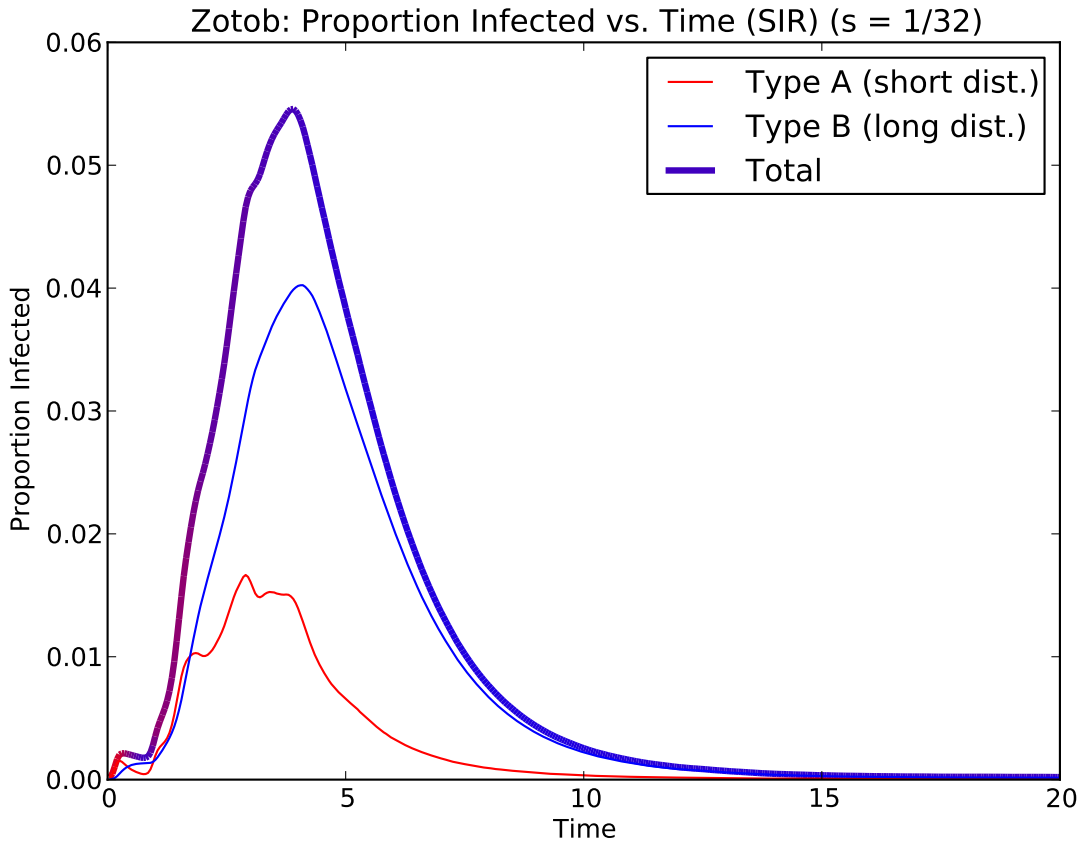


Figure 3.16: Plot of a Zotob infection versus time with $s = 1/32$. The thick line is the sum of the Type A and Type B infections, and is colored according to the proportions of Type A (red, short-distance) and Type B (blue, long-distance) worms making up the total infection. Results averaged over 10 Zotob simulations.

upon highly susceptible households, in which case those employing mainly short-distance dispersal perform exceptionally well, worms using too much short-distance dispersal, such as the Zotob worms with $s = 1/512$, will have a great deal of difficulty infecting new hosts before the infectious individuals recover permanently.

Zotob worms with $s = 1/32$, on the other hand, perform well when simulated using the SIR model for just the opposite reason. The course of such a simulation is provided in Figure 3.16. By employing more long-distance dispersal, these worms have more opportunities to find new areas of susceptibility before they have exhausted all the once-susceptible hosts in their current household. Zotob worms with $s = 1/32$ can spread to many machines via the

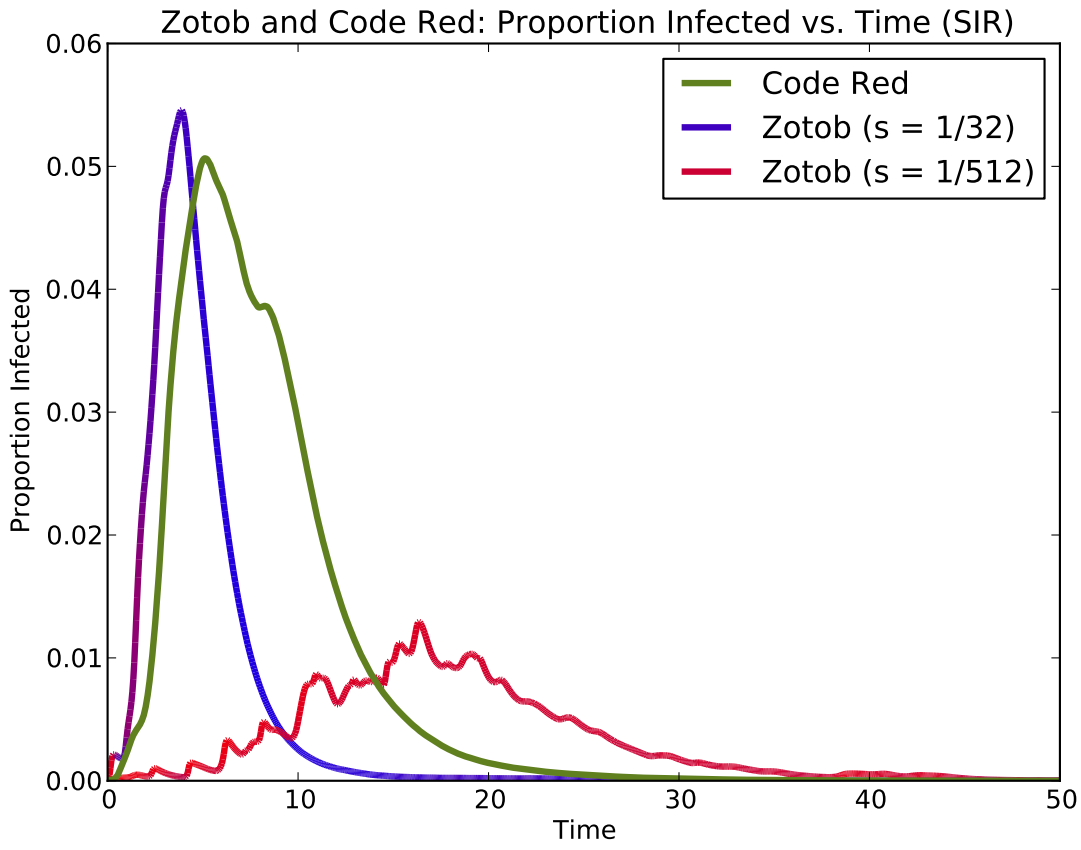


Figure 3.17: Plot of Code Red and two types of Zotob infections versus time. Results averaged over 10 simulations.

proliferation of some short-distance dispersal while benefiting from the stability of many long-distance infections seeking out new susceptible areas of the Internet. Indeed, likely because of this stability, Zotob with $s = 1/32$ performs even better than Code Red when simulated using the SIR model, as depicted in Figure 3.17. Zotob with $s = 1/32$ spreads not only more quickly but also to more machines at its peak than Code Red or Zotob with $s = 1/512$. Long-distance dispersal thus appears to be an important trait for worms spreading in situations where recovery means permanent removal from the population of susceptible individuals. Code Red likely suffers from a less extreme version of the spiky behavior of Zotob with $s = 1/512$, successfully infecting new areas, but dying out before another vulnerable pocket of the Internet can be found due to a lack of long-distance dispersal. Significant medium-distance dispersal likely contributes to Code Red's success compared to Zotob with $s = 1/512$.

Chapter 4

CONCLUSION

We have found that preference-scanning worms exploit the emergent structure of the IPv4 Internet, spatial clustering of similar software and versions, and thus similar vulnerabilities. Due to this clustering of vulnerable hosts, short-distance infection attempts should be much more likely to find new hosts to invade. Nevertheless, if short-distance movement were the sole strategy, only a small portion of the Internet would become infected, namely those subnets containing the initial distribution of the worm. Some proportion of infection attempts must take place between more distant machines in order to colonize new regions of Internet address space, along with local dispersal to rapidly infect those newly colonized regions.

Additionally, fast dispersal, as measured by the initial growth rate of an infection, and the extent of a worm's infiltration, as measured by the peak proportion of susceptible hosts infected, are based on different distributions of dispersal distances. Successful dispersal speed depends strongly on a mixture of short- and medium- or long-distance infections, where medium-distance infections seem to give the worms an extra edge; worms employing a very high proportion of short-distance dispersal actually tend to spread more slowly. The extent of an infection, however, does seem correlate directly with the amount of short-distance infections that the worm performs, with more short-distance dispersal corresponding to a higher peak level of infection. Knowledge of these characteristics might be helpful in

analyzing the intent of worms in the wild. Zotob, for example, whose initial infections were 100% short-distance, might have been written with the goal of extremely fast proliferation, whereas the creators of Code Red, which utilized short-, medium- and long-distance dispersal throughout its lifetime, might have prioritized the infection of many hosts over a long period of time.

Our analysis of the differences between the SIR epidemiological model, in which recovered hosts can never become infected again, and the SIRS model, in which recovered hosts may become susceptible and thus infected any number of times, demonstrates the importance of computer literacy when it comes to recovering from a worm infection.

Our simulated worms were markedly more successful when we used the SIRS model, which corresponds to infected users removing a worm but not updating their software, than SIR, in which users both remove the infection and update the affected software to prevent subsequent infections by the same worm. To help dampen the spread of worms, Internet users should be aware of the important difference between cleansing their machine of the worm, perhaps by using a malware removal tool, and updating the susceptible software on their machine.

Internet worms are a serious plague to all users of the Internet, but they can and should be stopped. Although most Internet worms' proliferation strategies are not extremely complex, insufficient work has been done to characterize the dynamics of their epidemics. Fortunately, we have begun to elucidate the inner workings of preference-scanning worms, which is the first step towards designing systems that can better prevent future invasions.

Albanese, D. J., Wiacek, M. J., Salter, C. M., & Six, J. A. (2004, June 18). *The Case for Using Layered Defenses to Stop Worms* (Tech. Rep.). Network Architecture and Applications Division of the Systems and Network Attack Center, U.S. National Security Agency. Retrieved April 20, 2012, from `http://www.nsa.gov/ia/_files/support/WORMPAPER.pdf`

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*(1), 47–97.

Avlonitis, M., Magkos, E., Stefanidakis, M., & Chrissikopoulos, V. (2007, June). A Spatial Stochastic Model for Worm Propagation: Scale Effects. *Journal in Computer Virology*, *3*(2), 87–92.

Balthrop, J., Forrest, S., Newman, M., & Williamson, M. M. (2004, April). Technological networks and the spread of computer viruses. *Science*, *304*, 527–529.

Bank, D. (2004, May). Computer worm is turning faster; installing security patches is now constant rush job against speedier invaders. *The Wall Street Journal*, B3.

Barbour, A. D. (1978). Macdonald's model and the transmission of bilharzia. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, *72*(1), 6–15.

Barthélemy, M., Barrat, A., Pastor-Satorras, R., & Vespignani, A. (2005). Dynamical patterns of epidemic outbreaks in complex heterogeneous networks. *Journal of Theoretical Biology*, *235*, 275–288.

Bellovin, S., Cheswick, B., & Keromytis, A. (2006). Worm propagation strategies in an IPv6 Internet. *Login*, *31*(1), 70–76.

Chen, Z., Gao, L., & Kwiat, K. (2003, March). Modeling the Spread of Active Worms. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*. San Francisco, CA. Available from `http://www.ieee-infocom.org/2003/papers/46_03.pdf`

Chen, Z., & Ji, C. (2005). Spatial-Temporal Modeling of Malware Propagation in Networks. *IEEE Transactions on Neural Networks*, *16*(5), 1291–1303.

Chen, Z., & Ji, C. (2007, May). Measuring Network-Aware Worm Spreading Ability. In *IEEE International Conference on Computer Communications (INFOCOM 2007)* (Vol. 26, pp. 116–124).

Cotton, M., & Vegoda, L. (2010, January). *Special Use IPv4 Addresses* (RFC No. 5735). Internet Engineering Task Force. Available from `http://tools.ietf.org/html/rfc5735`

Daley, D., & Gani, J. (1994). A Deterministic General Epidemic Model in a Stratified Population. In F. Kelly (Ed.), *Probability, Statistics, and Optimisation* (pp. 117–132). Chichester, UK: John Wiley & Sons, Ltd.

Doyle, J. C., Alderson, D. L., Li, L., Low, S., Roughan, M., Shalunov, S., et al. (2005, October). The "Robust Yet Fragile" Nature of the Internet. *Proceedings of the National Academies of Science*, *102*(41), 14497–14502.

Duryea, M., Caraco, T., Gardner, G., Maniatty, W., & Szymanski, B. K. (1999). Population dispersion and equilibrium infection frequency in a spatial epidemic. *Physical Review D*, *132*, 511–519.

Dye, C., & Hasibeder, G. (1986). Population dynamics of mosquito-borne disease: Effects of flies which bite some people more frequently than others. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, *80*, 69–77.

Eichin, M. W., & Rochlis, J. A. (1989, May). With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. In *Proceedings of the 1989 IEEE Symposium of Research in Security and Privacy.* Oakland, CA.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., et al. (1999, June). *Hypertext Transfer Protocol – HTTP/1.1* (RFC No. 2616). The Internet Society. Available from `http://tools.ietf.org/html/rfc2616`

Filipe, J., & Gibson, G. (1998). Studying and approximating spatio-temporal models for epidemic spread and control. *Phil. Trans. R. Soc. Lond. B*, *353*, 2153–2162.

Filipe, J., & Maule, M. (2004). Effects of dispersal mechanisms on spatio-temporal development of epidemics. *Journal of Theoretical Biology*, *226*, 125–141.

Grabowski, A., & Kosiński, R. (2004). Epidemic spreading in a hierarchical social network. *Physical Review E*, *70*, art. no. 031908.

Hiebeler, D. E. (2000). Populations on fragmented landscapes with spatially structured heterogeneities: Landscape generation and local dispersal. *Ecology*, *81*(6), 1629–1641.

Hiebeler, D. E. (2004). Competition between near and far dispersers in spatially structured habitats. *Theoretical Population Biology*, *66*(3), 205–218.

Hiebeler, D. E. (2006). Moment equations and dynamics of a household SIS epidemiological model. *Bulletin of Mathematical Biology*, *68*(6), 1315–1333.

Hiebeler, D. E., Michaud, I. J., Ackerman, H. H., Iosevich, S. R., & Robinson, A. (2011). Multigeneration reproduction ratios and the effects of clustered unvaccinated individuals on epidemic outbreak. *Bulletin of Mathematical Biology*, *73*(12), 3047–3070.

Hwang, D.-U., Boccaletti, S., Moreno, Y., & López-Ruiz, R. (2005, April). Thresholds for epidemic outbreaks in finite scale-free networks. *Mathematical Biosciences and Engineering*, *2*(2), 317–327.

Keeling, M. (2005). The implications of network structure for epidemic dynamics. *Theoretical Population Biology*, *67*, 1–8.

Kephart, J. O., Chess, D. M., & White, S. R. (1993, May). Computers and Epidemiology. *IEEE Spectrum*, 20–26.

Kephart, J. O., & White, S. R. (1991). Directed-graph epidemiological models of computer viruses. In *IEEE Computer Society Symposium on Research in Security and Privacy* (pp. 343–359).

Kephart, J. O., & White, S. R. (1993, May). Measuring and modeling computer virus prevalence. In *IEEE Symposium on Research in Security and Privacy* (pp. 2–15).

Kotliar, N. B., & Wiens, J. A. (1990). Multiple scales of patchiness and patch structure: A hierarchical framework for the study of heterogeneity. *Oikos*, *59*, 253–260.

Magee, M. (2007, February). *W32.Zotob.E: Technical Details.* Symantec Corporation. Retrieved April 15, 2012, from `http://www.symantec.com/security_response/writeup.jsp?docid=2005-081615-4443-99&tabid=2`

Moore, C., & Newman, M. (2000). Epidemics and percolation in small-world networks. *Physical Review E*, *61*(5), 5678–5682.

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., & Weaver, N. (2003, July). Inside the Slammer Worm. *IEEE Security and Privacy*, *1*(4), 33–39.

Moore, D., & Shannon, C. (2001). *The Spread of the Code-Red Worm (CRv2).* The Cooperative Association for Internet Data Analysis (CAIDA). Retrieved April 15, 2012, from `http://www.caida.org/research/security/code-red/coderedv2_analysis.xml`

Moore, D., Shannon, C., & Brown, J. (2002). Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *Proceedings of the Second ACM SIGCOMM Workshop on Internet Measurement* (pp. 273–284).

Moore, D., Shannon, C., Voelker, G., & Savage, S. (2003). Internet Quarantine: Requirements for Containing Self-Propagating Code. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)* (pp. 1901–1910). San Francisco, CA.

Nazario, J. (2004). *Defense and detection strategies against Internet worms*. Boston, MA: Artech House.

Newman, M. (2002). The spread of epidemic disease on networks. *Phys. Rev. E*, *66*, art. no. 016128.

Pastor-Satorras, R., & Vespignani, A. (2001). Epidemic dynamics and endemic states in complex networks. *Physical Review E*, *63*, art. no. 066117.

Pastor-Satorras, R., & Vespignani, A. (2004). Epidemic Spreading in Scale-Free Networks. In *Evolution and Structure of the Internet* (pp. 180–210). Cambridge, United Kingdom: Cambridge University Press.

Richtel, M. (2005, August 17). Virus Attacks Windows Computers at Companies. *The New York Times*, sec. D2. Available from http://www.nytimes.com/2005/08/17/technology/17virus.html

RIPE NCC. (2011, July). *Statistics: IPv6 networks over time.* (RIPE Network Coordination Centre (NCC); http://www.ipv6actnow.org/info/statistics/)

Staniford, S., Paxson, V., & Weaver, N. (2002). How to 0wn the Internet in Your Spare Time. In *Proceedings of the 11th USENIX Security Symposium.*

Vogt, T. (2004, February 16). *Simulating and Optimising Worm Propagation Algorithms.* Available from http://web.lemuria.org/security/WormPropagation.pdf

Wang, C., Knight, J. C., & Elder, M. (2000). On computer viral infection and the effect of immunization. In *IEEE Computer Security Applications Conference (ACSAC '00)* (Vol. 16, pp. 246–256).

Weaver, N. (2002, March). *Potential Strategies for High Speed Active Worms: A Worst Case Analysis.* University of California, Berkeley. Available from www.cgisecurity.com/lib/worms.pdf

Wikipedia contributors. (2012, February 24). Reserved IP addresses. In *Wikipedia, the free encyclopedia.* Retrieved April 15, 2012, from http://en.wikipedia.org/wiki/Reserved_IP_addresses

Zou, C. C., Towsley, D., & Gong, W. (2004, October). Email Worm Modeling and Defense. In *13th International Conference on Computer Communications and Networks (ICCCN'04).* Chicago, IL.

Zou, C. C., Towsley, D., & Gong, W. (2006, July). On the Performance of Internet Worm Scanning Strategies. *Performance Evaluation*, *63*(7), 700–723.

Author's Biography

Emma Taylor Strubell was born in Las Cruces, New Mexico on June 22, 1989. She was raised in Cape Elizabeth, Maine, and attended high school at The Taft School in Watertown, Connecticut, spending her junior year in Rennes, France. Emma is a computer science major with a minor in mathematics. She is a member of the Upsilon Pi Epsilon, Pi Mu Epsilon, and Phi Beta Kappa honor societies. After graduation, Emma plans to pursue a Ph.D. in computer science at the University of Massachusetts, Amherst.