Dynamic Quarantine of Internet Worms

Cynthia Wong, Chenxi Wang[1], Dawn Song, Stan Bielski, Gregory R. Ganger[2]

CMU-PDL-03-108
Dec. 2003

Parallel Data Laboratory
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

If we limit the contact rate of worm traffic, can we alleviate and ultimately contain Internet worms? This paper sets out to answer this question. Specifically, we are interested in analyzing different deployment strategies of rate control mechanisms and the effect thereof on suppressing the spread of worm code. We use both analytical models and simulation experiments. We find that rate control at individual hosts or edge routers yields a slowdown that is linear in the number of hosts (or routers) with the rate limiting filters. Limiting contact rate at the backbone routers, however, is substantially more effective—it renders a slowdown comparable to deploying rate limiting filters at every individual host that is covered. This result holds true even when susceptible and infected hosts are patched and immunized dynamically. To provide context for our analysis, we examine real traffic traces obtained from a campus computing network. We observe that rate throttling could be enforced with minimal impact on legitimate communications. Two worms observed in the traces, however, would be significantly slowed down.

[1] This work was partially supported by the National Science Foundation under Grant No. CCR-0208853 and ANI-0326472.

[2] This research is partially sponsored by the Air Force Research Laboratory, under agreement number F49620-01-1-0433.

Acknowledgements: We thank the members and companies of the PDL Consortium (including EMC, Hewlett-Packard, Hitachi, IBM, Intel, Microsoft, Network Appliance, Oracle, Panasas, Seagate, Sun, and Veritas) for their support.

Keywords: Internet worms, modeling, worm throttling, rate limiting

1 Introduction

Since the original “Internet worm” [3] in 1988, computer worms continue to wreak havoc on the Internet. The recent SQL Slammer worm infected over 90% of the vulnerable hosts on the Internet within ten minutes [10]. With such voracity, the manual patch-’em-as-they-go approach simply does not work. We need automated detection and response to defend against worm outbreaks.

One class of techniques that seems promising is rate control—schemes that aim to limit the contact rate of worm traffic [5, 17]. Since worms typically spread at a rapid speed from host to host, restricting the contact rate of a worm constrains how fast the infection can spread in the network. Previous proposals of rate control consider deploying such mechanisms primarily at the individual host level. In this paper, we investigate rate control at individual end hosts and at the *edge* and *backbone* routers, for both random propagation and local-preferential connection worms. Our analysis shows that both host and edge-router based rate control result in a slowdown (in the spreading rate of the worm) that is linear to the number of hosts (routers) implementing the rate limiting filter. In particular, host-based rate control has very little benefit unless rate limiting filters are universally deployed. Rate control at the backbone routers, however, is substantially more effective. Our results hold true for both random propagation worms (e.g., Code Red I) and worms that spread via a preferential connection algorithm such as those that target local hosts within a subnet.

Results are similar when dynamic immunization is taken into account. As the worm spreads and the knowledge of the worm disseminates, an increasing number of hosts (both infected and susceptible) will be patched, immunized and consequently removed from the susceptible population. In an effort to study realistic worm attacks, the models in this paper incorporate dynamically changing the immunization rates. This is in contrast to the traditional models for which the rate of immunization remains constant throughout the infection outbreak [7, 16, 2, 6, 15].

To provide context for the models, we examine traffic traces obtained from a sizable campus computing network. We observe that limiting the rate of unique IP addresses contacted (as in [17]) from the edge of the departmental network to no more than 16 (total contacts) per five-second period would almost never affect legitimate traffic. Individual host rates can be kept to under four per five-second period. Limiting only non-DNS-translated IP address contacts [5] can reduce the contact rate by another factor of 2 – 4. Our traces also captured the behavior of machines infected by two worms: Welchia and Blaster. The results confirm that infected machines exhibit much higher contact rates and could be dramatically slowed by rate limiting.

Combining practical rate limits with our models allows us to estimate how well such approaches might work in practice. For instance, to secure an enterprise network from worms that propagate using a local-preferential connection algorithm, our study shows that unless rate limiting filters are deployed at both the edge routers and a certain percentage of the end hosts, little benefit will be gained.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 gives a brief background in epidemiological models. Sections 4 and 5 study deployment strategies of rate limiting schemes. Section 6 incorporates dynamic immunization with rate control, and Section 7 presents a case study of real network traces. We summarize in section 8.

2 Related work

Several documented studies investigated computer worms and the ways in which they propagate. Staniford et al. presented a study of different types of worms and how they can cause damage on the Internet [13]. Zou et al. [19] analyzed the propagation of the Code Red worm and presented an analytic model for worm propagation; Moore et al. [10] analyzed the propagation of the Slammer worm and its effect on the Internet. These studies have not analyzed defense mechanisms in great depth.

Moore et al. [11] explored the design space for worm containment systems. They studied the efficacy of address blacklisting and content filtering with various deployment scenarios. They concluded that detection and containment must be initiated within minutes for such systems to be effective. Singh et al. [12] proposed a system for real-time detection of unknown worms using traffic analysis and content signatures. Zou et al. [18] proposed to monitor unused address space on ingress and egress routers to detect worms at their early propagation stage.

Our work differs from previous works in that we focus on analysis of rate control. As we demonstrate in Sections 5 and 6, rate control mechanisms can be extremely effective in curtailing worm spread if deployed correctly.

The primary contribution of our work is the analysis of different deployment strategies for rate control mechanisms. Williamson [17] proposed the idea of host-based rate limiting by restricting the number of new outgoing connections. Ganger et al. [5] proposed a scheme that analyzes and limits network traffic based on abnormal DNS lookup patterns. Both of these schemes are host based and did not explore other deployment options.

3 Background—epidemiological models

In this section we briefly introduce one class of epidemiological models, namely homogeneous models. Homogeneous models are widely used in the studies of human infections. A homogeneous model assumes homogeneous mixing among the individuals in the population [1, 8]; that is, every individual has equal contact to every one else in the population. This assumption is similar to the ways in which random propagation worms spread in computer networks. This model is described in more detail in [1]. A homogeneous model assumes a connected network with N nodes. It also assumes an average infection rate β across all links. If we represent total number of infected nodes at time t as I_t, a deterministic time evolution of I (infected hosts) can be obtained as below,

$$\frac{dI_t}{dt} = \beta I_t(N - I_t/N) \tag{1}$$

The solution to Equation (1) is $I/N = \frac{e^{\beta t}}{c+e^{\beta t}}$, where c is a constant. c is determined by the initial infection level. $c \to N-1$ when the initial infection level is low, since the fraction of infected hosts will be small.

From this we can see that the infection grows exponentially initially and reaches saturation after a certain point. The time takes to reach a certain infection level α is

$$t \doteq ln\alpha/\beta \tag{2}$$

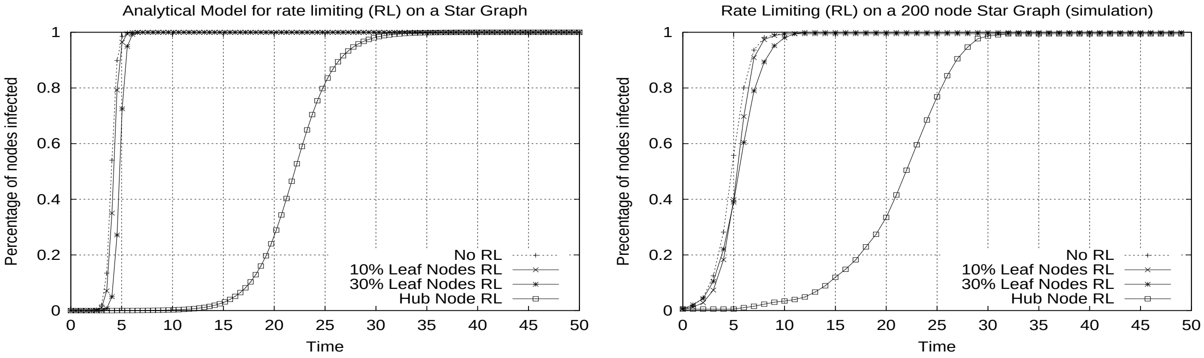
The analytical models described in the later parts of this paper are derived from the basic homogeneous model and share the same assumptions.

4 Rate Limiting

In this section, we present a study on rate limiting mechanisms as a defense to combat the propagation of Internet worms. Rate limiting is a mechanism by which an element in the network can restrict the rate of communication with other elements. Since worms spread rapidly via fast connections to uninfected machines, rate limiting can help suppress the propagation of the worm. A number of rate limiting schemes have been proposed in the literature, including Williamson's virus throttle [17] and Ganger's DNS based scheme [5]. However, it is not known precisely how and where rate control mechanisms should be deployed in a network. Clearly, instrumenting rate control on every individual node in a network is expensive administratively and hence not feasible. The question then becomes: are there alternative deployment strategies that can yield a more desirable effect than others?

We believe that the answer to this question is yes. In this section we illustrate the effect of different deployment strategies using a star graph topology. Consider a star graph where a central hub node is connected to all the leaf nodes. We analyze two deployment scenarios: a) rate control at a certain percentage of the leaf nodes, and b) rate control at the center hub node only. Note that a star topology is very different from the Internet's topology and the study of a star topology is mainly for demonstration of the difference from deployment at leaf and hub nodes.

Figure 1: Plots showing the differences between various rate-limiting deployment mechanisms on a 200-node star topology



(a) Analytical model of infection evolution

(b) Simulated infection evolution

Deployment at leaf nodes: Assume we deploy rate limiting filters at q percent of the leaf nodes. Let $x_1 = I(1-q)$ be the number of infected nodes that are not confined by the filtering mechanism, $x_2 = Iq$ the number of infected nodes with the filter mechanism, β_1 the contact rate of the infected host without the filter, and β_2 the contact rate allowed by the filter, with $\beta_1 >> \beta_2$.

We obtain the time evolution of the infection as below,

$$\frac{dI}{dt} = x_1\beta_1(N-I)/N + x_2\beta_2(N-I)/N \tag{3}$$

Solving Equation (3) gives us $I/N = \frac{e^{\lambda t}}{c+e^{\lambda t}}$, where $\lambda = q\beta_2 + (1-q)\beta_1$. When $\beta_1 >> \beta_2$ and $e^{\lambda t}$ is small, $\lambda \doteq \beta_1(1-q)$. From this, we can derive that the time t to reach a certain infection level α is $t = ln\alpha/(\beta_1(1-q))$. That is, the rate of infection is proportional to $1-q$, the percentage of nodes that do not have rate limiting filters. Comparing to Equation (2), we can see that deploying rate limiting filters at the leaf nodes yields a linear slowdown that is proportional to the number of nodes that have rate control.

Deployment at hub: When deploying rate control at the center hub node, we need to consider both node-level and link-level rate limiting. Assume we deploy rate limiting at the hub node with rate β and link rate limiting with rate γ. When $\beta \geq \gamma I$ — the contact rate at the hub node is greater than the combined contact rate of infected leaf nodes — then γI is the primary limiting factor for infection propagation. Otherwise, the propagation is limited by the hub node contact rate, β.

For link-level rate limiting (this is when the contact rate at the hub node is higher than the combined contact rates of all the infected leaf nodes), we have

$$\frac{dI}{dt} = \gamma I(N-I)/N, \text{ when } \gamma I \leq \beta \tag{4}$$

Solving Equation (4) gives us

$$I/N = \frac{e^{\gamma t}}{c+e^{\gamma t}}, \text{ when } \gamma I \leq \beta$$

For node rate limiting (this is when the combined contact rates of the infected leaf nodes exceeds the hub node contact rate), we have

$$\frac{dI}{dt} = \beta(N-I)/N, \text{ when } \gamma I > \beta \tag{5}$$

Solving Equation (5) gives us

$$I/N = 1 - ce^{-\beta t/N}, \text{ when } \gamma I > \beta$$

From the solution to Equation (4), we can derive that the time t to reach an infection level α is $t \doteq N(ln(\alpha))/\beta$. Compared to rate control at the leaf nodes, this suggests a slowdown that is comparable to installing rate control filters at all of the leaf nodes — in which case $t = ln(\alpha)/\beta_2$. Indeed, the graph in Figure 1(a), which plots both leaf-node and hub-node rate control on a 200-node star topology, indicates exactly that. Figure 1(b) shows simulated propagations on the same topology.

In our simulation, we limited the links to 10 packets per second with the hub rate limit $\beta = 0.01$. The simulation results are an average of ten simulation runs. For leaf-node rate control, we simulated rate limiting at 10% and 30% of the leaf nodes. As shown in Figure 1(b), rate limiting at 10% of the leaf nodes has neligible impact. Rate limiting at 30% of the leaf nodes results in a slight slowdown of the infection rate. Rate control at the hub node is significantly more effective. For instance, reaching a level of 60% infection with rate limiting at 30% of the leaf nodes is approximately three times quicker than rate limiting at the hub. These results confirm our analytical model.

This simple but illustrative example shows that deployment strategies have a significant impact on the effectiveness of rate control schemes. On the Internet, we can deploy rate control at end hosts, edge routers, and backbone routers. In the next section we investigate each of these deployment cases.

5 Deploying rate control on the Internet

In this section we investigate three different ways of deploying rate limiting schemes on the Internet: on individual hosts, edge-routers, and at backbone routers. We develop a mathematical model to reason about each deployment strategy's effectiveness, and conduct simulation experiments to confirm the model's predictions.

5.1 Host-based rate limiting

Deploying rate limiting filters at individual hosts is similar to rate limiting at the leaf nodes of a star topology as described in Section 4. Again, let q be the percentage of nodes that install the filter mechanism. $x_1 = I(1-q)$ is the number of infected nodes that are not confined by the filter mechanism, and $x_2 = Iq$ is the number of infected nodes with the filter mechanism. β_1 is the contact rate of the infected host without the filter, β_2 is the contact rate allowed by the filter, and $\beta_1 >> \beta_2$.

Similarly, we can use Equation (3) to model the time evolution of infection. The solution to Equation (3) gives us

$$I/N = \frac{e^{\lambda t}}{C + e^{\lambda t}}, \text{ where } \lambda = q\beta_2 + (1-q)\beta_1$$

When $\beta_1 >> \beta_2$, $\lambda \doteq \beta_1(1-q)$. The analysis in Section 4 on rate limiting on leaf nodes also holds here. Figure 2 shows the time evolution of I with β_1 = 0.8 and β_2 = 0.01. As we see in Figure 2, the deployment of host-based confinement mechanisms yields a linear slowdown in the infection rate of the worm. Note the difference between 80% deployment and 100% deployment of rate limiting, this shows that rate limiting has very little benefit unless all end hosts implement rate limiting.

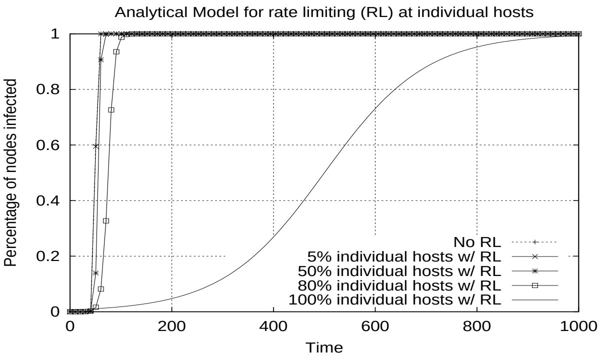


Figure 2: Analytical model for rate limiting at individual hosts with β_1 = 0.8 and β_2 = 0.01

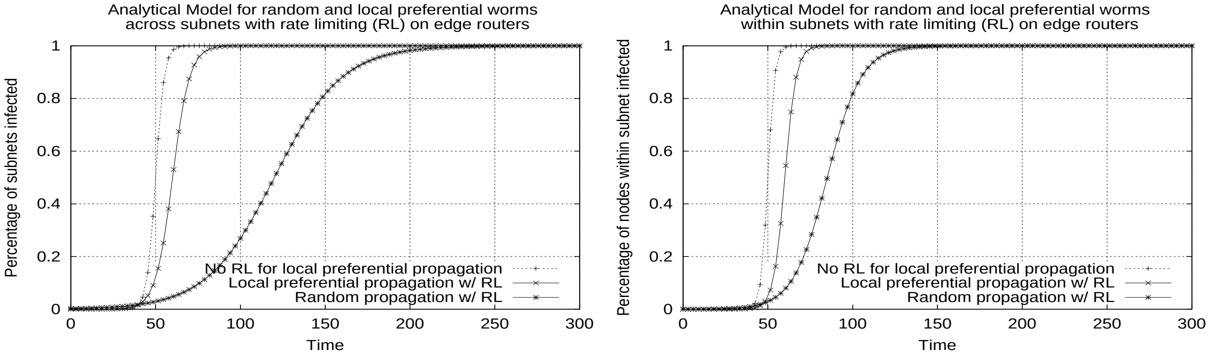
5.2 Rate limiting at edge routers

Edge-router based deployment is similar to the host-based rate limiting scheme. From the set of networks that install the filter, we can calculate the effective *q* (percentage of nodes that install the filter mechanism) and the rest of the calculation is the same.

When filters are installed at edge routers, worms propagate much faster within the subnet than across the Internet. We denote the contact rate within the subnet as β_1 and the contact rate across the Internet as β_2. Clearly, $\beta_1 \geq \beta_2$. For a random propagation worm, the infection growth within the subnet has the form $x = \frac{e^{\beta_1 t}}{C_1 + e^{\beta_1 t}}$, where x is the number of infected nodes within a particular subnet. The number of subnets infected has a similar growth form $y = \frac{e^{\beta_2 t}}{C_2 + e^{\beta_2 t}}$, where y is the number of infected subnets.

For worms that use a preferential targeting algorithm (i.e., those that target nodes within the same subnet), the growth formula stays the same except for that the infection rate within the subnet, β_1, could be substantially larger than that of a random propagating worm. Consequently, the effectiveness of rate control at edge routers diminishes when a worm employs an intelligent targeting algorithm such as subnet preferential selection.

Figure 3: Analytical models for random and local preferential worms



(a) Spread of worm across subnets

(b) Spread of worm within a subnet

Figure 3 depicts the analytical models for both local preferential connection and random propagation worms with rate limiting filters at the edge routers. It shows the time evolution of the percentage of hosts infected with β_1 = 0.8 and β_2 = 0.01. In the base case with no rate limiting, the infection grows exponentially before it reaches its maximum limit. With rate control there is a slight slowdown in the rate of infection. As shown in Figure 3(*a*), our model indicates that edge router rate limiting is more effective for the random propagation model. To verify this, we created simulations to compare edge router rate limiting for both local preferential and random propagation models. The results of the simulations are shown in Section 5.4

5.3 Rate limiting at backbone routers

In this section we investigate rate limiting at the backbone routers of the Internet. In order for a worm to propagate from one network to another, the worm packets need to go through backbone routers on the Internet. Therefore, deploying rate limiting mechanisms at the backbone routers can help throttling worm propagation. We perform an approximate analysis of rate limiting at backbone routers below.

If we deploy the rate limiting mechanism on the core routers that cover α percent of the total IP-to-IP paths, then

$$\frac{dI}{dt} = I\beta(1-\alpha)(N-I)/N + \delta(N-I)/N, \tag{6}$$

where β is the contact rate of one infected host, $\delta = \min(I\beta\alpha, rN/2^{32})$, and r is the average overall allowable rate of the routers with the rate limiting control. When r is relatively small, the right hand side of Equation 6 can be approximated by only the first term. We can thus obtain $I/N = \frac{e^{\lambda t}}{c+e^{\lambda t}}$, where $\lambda = \beta(1-\alpha)$ and c is a constant.

5.4 Simulation Results

The simulator that is used to conduct our experiments is built on top of Network Simulator (ns-2) [4]. All experiments in this section are conducted using an 1,000 node power-law graph generated by BRITE [9]. The graph shares similar characteristics to an AS topology such as the Oregon router views. Unless specified otherwise, each simulation is averaged over 10 individual runs. In addition, the time units in all our simulations are simulation ticks as defined by ns-2.

We begin each simulation with a random set of initial infections. At each time unit each infected node will attempt to infect everyone else with infection probability β. The infection packet is routed using a shortest path algorithm through the network. Links that have the rate limiting mechanism will only route packets at a rate of γ.

In order to experiment with the different deployment cases, we designate the top 5% and 10% of nodes with the most number of connections as backbone and edge routers respectively. The remaining nodes are end hosts. Rate limiting is implemented by restricting the maximal number of packets each link can route at each time tick and queuing the remaining packets. In order to ensure that normal traffic gets routed, we assign each rate-controlled link a base communication rate of 10 packets per second. We then compute a link weight that is proportional to the number of routing table entries the link occupies. We multiply this weight to the base rate to obtain the actual link rate simulated for each link. We believe that this simulated routing will allow most normal traffic to be routed through since the most utilized links will have a higher throughput.

Figure 4 shows the simulation results for random propagation worms, for the cases of no rate limiting, rate limiting at 5% of the end hosts, edge routers and backbone routers. As shown, the simulation results confirm our analytical models in Sections 5.1 and 5.2. More specifically, there is negligible difference between no rate limiting and rate limiting at 5% of end hosts. While rate limiting at the edge routers shows a slight improvement, rate limiting at the backbone routers renders a substantial improvement. Compared to the case of end host and edge router based rate limiting, it takes approximately five times as long for the worm to spread to 50% of all susceptible hosts if rate limiting is implemented at the backbone routers.

Figure 5 shows the simulated propagations for rate limiting at the edge router for both local preferential and random propagation worms within subnets. The dotted lines are the base cases (with no rate limiting) for

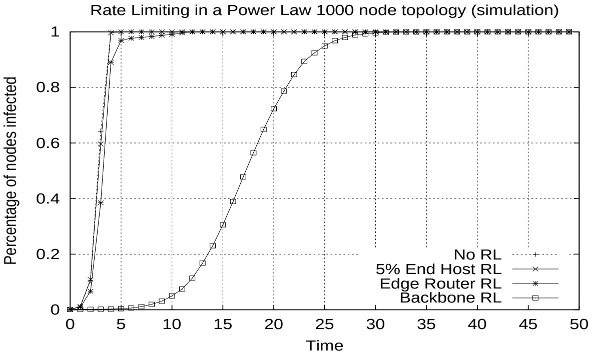


Figure 4: Simulation of rate limiting at end hosts, edge routers and backbone routers.

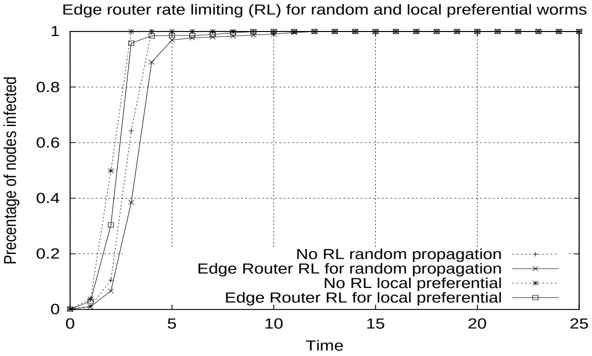


Figure 5: Simulation and comparison of rate limiting within subnets at the edge router for local preferential and random propagation worms.

local preferential and random worms respectively. As our simulations show, there is very little perceivable benefits for implementing rate limiting at the edge routers if worms propagate using a local preferential algorithm. For random propagation worms, however, rate control at the edge routers still yields a 50% slowdown. Clearly, edge router based rate limiting is more effective in suppressing random propagation worms as opposed to worms that propagate via local preferential connections. These results also confirm our analytical model described in Section 5.2. Figure 6 shows the simulated propagation for local preferential worms for both host- and backbone-router-based rate limiting across subnets. As shown, even with a 30% deployment of rate limiting mechanisms at the end hosts there is negligible difference when compared to no rate limiting. Deploying rate limiting filters on the backbone routers, as shown in Figure 6, is substantially

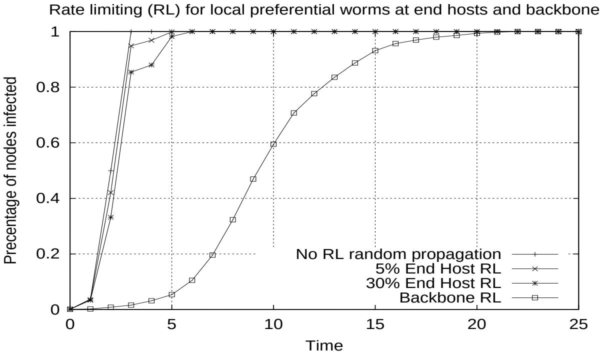


Figure 6: Simulation of rate limiting across subnets for local preferential worms at the end hosts and backbone routers.

more effective.

6 The effect of dynamic immunization

Thus far in our analysis we have ignored the effect of patching and dynamic immunization. Clearly, the infection progress will be hampered when the exploited vulnerabilities are patched (immunized) dynamically. In this section, we examine dynamic immunization and its effect on rate limiting.

6.1 Delayed Immunization

The immunization model we consider here assumes that the immunization process starts at time d, after a certain percentage of hosts are infected. Thereafter, in each time interval, each susceptible host will be patched with probability μ. The following differential equations model the dynamics of the worm propagation in the presence of immunization:

$$\frac{dI}{dt} = I\beta\frac{N-I}{N}, \text{ when } t \leq d$$

$$\frac{dI}{dt} = I\beta\frac{N-I}{N} - I\mu, \text{ when } t > d$$

$$\frac{dN}{dt} = -\mu N, \text{ when } t > d$$

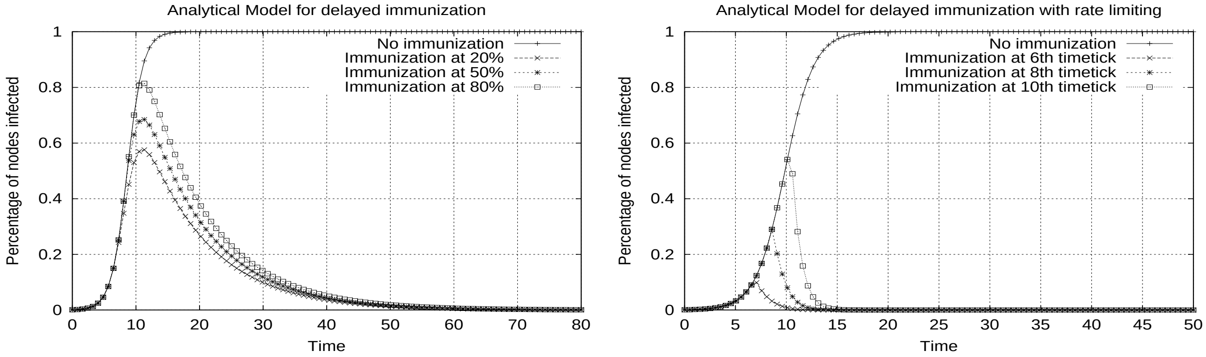
Solving the above equations gives us,

$$\frac{I}{N_0} = \frac{e^{\beta t}}{c + e^{\beta t}} \text{ when } t \leq d$$

$$\frac{I}{N_0} = \frac{e^{(\beta-\mu)(t-d)}}{c_0 + e^{\beta(t-d)}}, \text{ when } t > d$$

where N_0 denotes the initial number of susceptible hosts.

Figure 7: Analytical Models (with and without rate limiting) for delayed immunization on a 1000-node power-law graph



(a) Analytical model of infection evolution for delayed immunization

(b) Analytical model of infection evolution for delayed immunization with rate limiting

Figure 7(a) shows a plot of the equations. We also conducted simulations of delayed immunization on a synthetic 1000-node power-law graph with $d = 20\%$, 50%, and 80% infection (nodes infected) with $\beta = 0.8$ and $\mu = 0.1$. The results are shown in Figure 8(a). Clearly, the earlier immunization takes place, the more effective it is. In Figure 8(a), immunization starting at 20% infection yielded a total infected population of 80% of the nodes, as opposed to 90% infected when immunizing at 50% and 98% infected at 80%.

We note that the assumption of a constant probability of immunization, denoted by μ, is not completely realistic. In reality, the probability of immunization may increase as the worm spreads and as the vulnerability it exploits becomes widely publicized. Similarly, the probability of immunization may decrease as the infection becomes a rarer occurrence, i.e., on its way to extinction. We believe that the rate of immunization observes a bell curve. However, the exact shape of such a curve is not easily obtainable and we lack data to confirm the rate of immunization. Therefore in this paper we use the simple assumption of immunization at a constant rate.

6.2 Rate control with dynamic immunization

In this section we examine the effect of delayed immunization with rate limiting. We focus on the case of rate limiting at backbone routers (since that is the most effective strategy according to the analysis in Section 5). Assuming the first instance of immunization occurs at time d, the growth of the infection with delayed immunization and rate control at the backbone routers is as follows:

$$\frac{dI}{dt} = I\beta(1-\alpha)(N-I)/N + \delta(N-I)/N, \text{when } t \leq d$$

$$\frac{dI}{dt} = I\beta(1-\alpha)(N-I)/N + \delta(N-I)/N - \mu I, \text{when } t > d$$

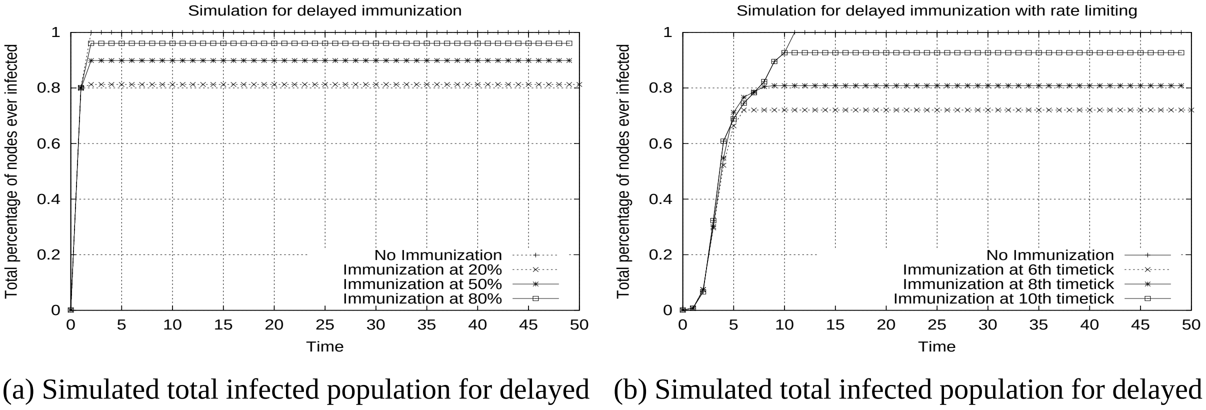
$$\frac{dN}{dt} = -\mu N, \text{when } t > d$$

where β is the contact rate of one infected host, $\delta = \min(I\beta\alpha, rN/2^{32})$, r is the average overall allowable rate of the routers with rate limiting control and α is the percentage of paths that have rate limited links. When r is relatively small, the solution can be approximated as

$$\frac{I}{N_0} = \frac{e^{\gamma t}}{c + e^{\gamma t}} \text{ when } t \leq d,$$

$$\frac{I}{N_0} = \frac{e^{(\gamma-\mu)(t-d)}}{c_0 + e^{\gamma(t-d)}}, \text{ when } t > d, \text{ where } \gamma = \beta(1-\alpha)$$

Figure 8: Simulations of delayed immunization (with and without rate limiting) on a 1000-node power-law graph



(a) Simulated total infected population for delayed immunization

(b) Simulated total infected population for delayed immunization with rate limiting

We also conducted simulations of delayed immunization with rate limiting on a synthetic 1000-node power-law graph with $\beta = 0.8$ and $\mu = 0.1$. Since the goal here is to identify the benefits of rate limiting, the timeticks chosen in both analytical and simulation are the timeticks at which immunization started in our analytical model for delayed immunization without rate limiting (e.g. for immunization starting at 20%, our analytical model shows that it should happen around the 6th timetick). Figure 7(b) plots the analytical model and Figure 8(b) plots the simulated results of delayed immunization with rate limiting. The plots show exactly how immunization delays in combination with rate limiting at the backbone affects the infection propagation. Recall that in Figure 8(a), immunization at 20% with no rate limiting results in a total infected population of 80%. Figure 8(b) shows a similar experiment with the same simulation parameters but with rate limiting, which results in a total infected population of 72%, a 10% drop from the case where no rate limiting was implemented. The same results hold for other values of delay period d. In summary, rate limiting helps to slow down the spread and as a result buys time for system administrators to patch their systems and ultimately minimize the damage of worm outbreaks.

7 Rate Limiting in Practice

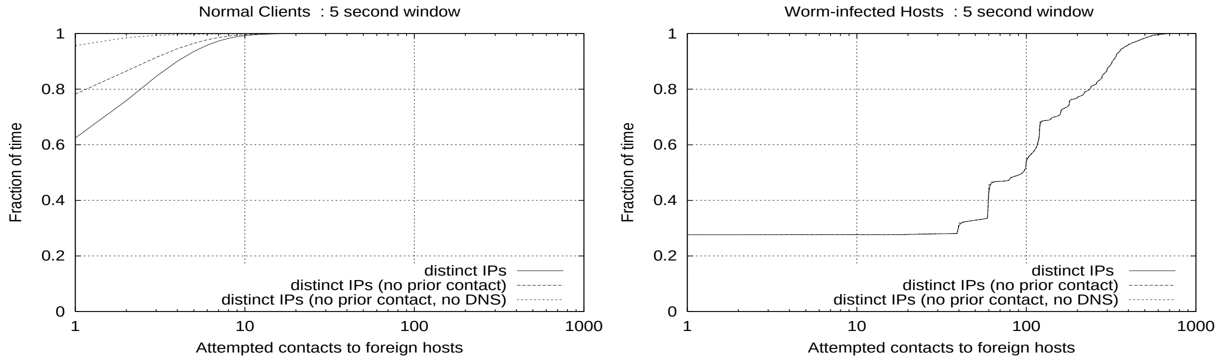
This section presents an analysis of real network traces, with a goal of identifying rates at which connections can be throttled in practice. Specifically, we wish to identify rate limits that an enterprise network can realistically implement that will significantly slow worms while having minimal impact on legitimate communications. Using these rates in our models produces a corresponding propagation prediction that might be viable in practice.

We focus on two recently proposed techniques for rate limiting. The first, proposed by Williamson [17], restricts the number of unique IP addresses with which a host communicates in a given period of time; the default discussed in that paper was five per second (per host). The second, proposed by Ganger et al. [5], restricts the number of unknown IP addresses (those without valid DNS cache entries and that did not initiate contact) to which a host can initiate connections in a given period of time; the default discussed was six per minute (per host). The second technique focuses on the common approach used by self-propagating worms to identify target hosts: picking pseudo-random 32-bit values to use as an IP address (thus performing no DNS translation).

We evaluated these techniques using a 23-day trace from the edge router for CMU's Electrical and Computer Engineering (ECE) Department. The traces recorded in an anonymized form all IP and common second layer headers of traffic (e.g., TCP or UDP) entering or exiting the ECE network from August 15th until September 7th, 2003. The contents of all DNS packets were recorded and anonymized. In addition to the regular activity of the department, this period includes two major worm outbreaks: Blaster [14] and Welchia.

Through examining the traces, we were able to partition the ECE subnet (1128 hosts total) into four types of hosts: normal "desktop" clients, servers, clients running peer-to-peer applications, and systems infected by worms. Each type of hosts exhibited significantly different connectivity characteristics. The 999 "Normal Clients" exhibited traffic patterns driven by client-server communication, such as HTTP, AFS, and FTP traffic. 17 "Servers" provide network services, such as SMTP, DNS, or IMAP / POP. The 33 clients running peer-to-peer applications (in these traces Kazaa, Gnutella, Bittorrent, and edonkey) were placed in their own category because they exhibit greater connectivity than normal hosts. This can be attributed to the nature of peer-to-peer systems; packets must be exchanged periodically in order to establish which hosts are on the network and the content they serve. Finally, 79 systems were observed to have been infected by the Blaster and/or Welchia worms. Both these worms exploited the Windows DCOM RPC vulnerability. Blaster scanned subnets for other vulnerable hosts by attempting to send itself to TCP destination port 135. Welchia was a "patching" worm which first scanned subnets for vulnerable hosts using ICMP ping requests. If a host replied, Welchia attempted to infect the system, make further attempts to propagate, patch the vulnerability, and reboot the host. We were able to differentiate between the two worms by looking for a large amount of ICMP echo requests intermixed with TCP SYNs to port 135. We found that although Welchia's intention was benign, its peak scanning rate was an order of magnitude greater than Blaster's.[*]

Figure 9: CDF of Contact rates in a five second interval for normal and infected clients



(a) CDF of aggregate contact rates for 999 "normal desktop" clients. Note how the contact classification refinements result in lower values.

(b) CDF of aggregate contact rates for 79 clients infected by the Blaster and/or Welchia worm.

Figure 9 shows the observed aggregate contact rates for (a) normal clients and (b) worm-infected clients. As shown, they are very different. In addition to the solid lines, which indicate the number of distinct IP addresses contacted in a 5-second period, two other lines are given to indicate the effect of possible refinements on rate limiting. The dashed line shows the number of distinct IP addresses contacted from within the network. The dotted line shows the number of distinct IP addresses contacted from within the network and not counting those for which valid DNS translations are obtained. Clearly, these refinements may be useful in limiting contact rates to lower numbers while having less impact on legitimate communications. For instance, to avoid having impact 99.9% of the time, inside-to-outside contact rate could be limited to 16 per five seconds for all contacts at the edge router, 14 per five seconds for contacts to hosts that did not initiate contact first, or 9 per five seconds for contacts to hosts for which a valid DNS translation did not exist or did not initiate contact first. The tightness of the three lines in the worm-infected graph support this statement, showing that worms traffic spike all three metrics.

The P2P and server systems are less well-behaved than normal systems and less ill-behaved (in terms of contact rate) than worm-infected systems. But, the contact rate limits would have to be greatly increased in order to avoid impacting regular traffic. Specifically for P2P clients, the network could be limited to 89 per five seconds for all contacts, 61 per five seconds for contacts to hosts that did not initiate contact first, or 26 per five seconds for contacts to hosts for which a valid DNS translation did not exist or did not initiate contact first. Alternately, an administrator could categorize systems as we have done, and give them distinct rate limits. This would tightly restrict most systems (those not pre-determined to be special), while allowing special others to contact at higher rates. Of course, performance penalties will be faced by new P2P users, until they convince the security administrator to deem them ”special”. Many administrators would prefer this model to the unconstrained load spikes that they currently face, and have to diagnose, as new P2P applications are introduced to their environment.

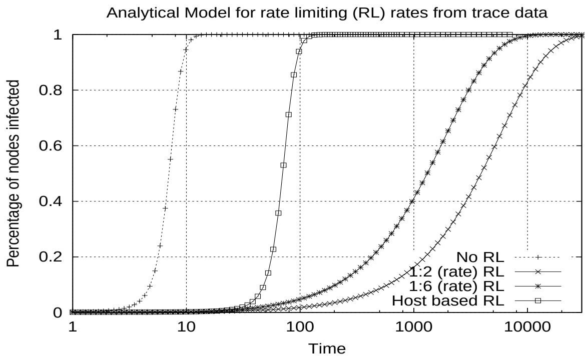
Rather than aggregate limits at the edge routers, as discussed above, another way to limit contact rates is per individual host (e.g., in host network stacks [17] on smart network cards or switches [5]). Our analysis of

[*] We discovered an instance of Welchia that scanned 7068 hosts in a minute. By contrast, Blaster's peak scanning rate was only 671 hosts in a minute. Blaster, however, was much more persistent in its propagation attempts.

the traces indicates that the resulting restrictions can safely limit a single "normal desktop" system initiating contact to, for example, four unique IP addresses per five seconds or one unique non-DNS-translated IP addresses per five seconds. Although these numbers are lower, however, the 1128 machines in the network could conceivably each use their full slot when a worm infects them, meaning that the aggregate contact rate from the intra-net would be much higher than the rate limits discussed for the edge router case. This suggests that per-host rate limits are a poor way to protect the external Internet from internal worm traffic.

Per-host limits, however, are a much better at (in fact, the only way) to protecting the internal network [5] once worms get past the outer firewall. Section 5 quantifies this benefit.

A final observation from the traces relates to the choice of a rate limit window size. We observed that longer windows accommodate lower long-term rate limits, because heavy-contact rates tend to be bursty. For example, for aggregate non-DNS rates, 99.9% of the values are five for one second, twelve for five seconds, and fifty for sixty seconds. The downside to a long window, however, is that one could face a lengthy delay after filling it, before the next connection is allowed. Visible disruptions of this sort may make long windows untenable in practice. One option worth exploring is hybrid windows with, for example, one short window to prevent long delays and one longer window to provide better rate-limiting. Figure 10 illustrates the effect of different rate limits on worm propagation. We approximate Williamson's IP throttling scheme and Ganger's DNS-based scheme using Equations (4) and (5) in Section 4. Although Equations (4) and (5) model deployment-at-hub, they can be used to approximate edge router rate limiting in the case of a single subnet. Recall β is the aggregated node contact rate while γ is the contact rate per link. As the traces indicated a lower aggregated rate for the DNS-based scheme, we choose the ratio of γ to β as 1:2 to represent the DNS-based scheme and the ratio of 1:6 for the IP throttling scheme. As shown, the rate limiting method based on DNS queries gives better results than the rate limiting method based purely on IP addresses visited. The plots also indicate unmistakably that aggregated rate limiting at the edge router performs better than per-host limits.



Note graph is plotted in log scale

Figure 10: Effect of rate limiting given the rates proposed by our trace study.

8 Conclusions

Recent work in rate limiting schemes such as traffic throttling [17] and secure NICs [5] show potential in mitigating widespread worm attacks. However, it is not known precisely how rate limiting filters should be deployed throughout a network and what a reasonable rate limit is in practice.

Our contributions in this work are twofold: First, we showed through modeling and simulation experiments that deploying rate limiting filters at the backbone routers is extremely effective. Rate control at the edge routers is helpful for randomly propagating worms, but does very little to suppress local preferential spreading worms. Individual host based rate control results in a slight linear slowdown of the worm spread, regardless of the spreading algorithm. A direct consequence of this analysis is that in order to secure an enterprise network, one must install rate limiting filters at the edge routers as well as some portion of the internal hosts.

Second, through a study of real network traces from a campus computing network, we discovered that there exist reasonable rate limits for an enterprise network that would severely restrict the spread of a worm but would have negligible impact on almost all legitimate traffic. This is especially encouraging since rate limiting filters can be easily installed and configured at various strategic points throughout a network. The result of the trace study confirmed that per-host rate limiting by itself is not sufficient to secure the enterprise network—aggregated rate limiting at the edge router must be employed at the same time to minimize the spread of worm attacks. This is the first study in this area of which we are aware of that has studied rate limiting with real traffic traces and has identified realistic rate limits in practice.

References

[1] Norman Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, London, 1975.

[2] Zesheng Chen, Lixin Gao, and Kevin Kwiat. Modeling the spread of active worms. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, April 2003.

[3] S. Eugene. The internet worm program: An analysis, 1988.

[4] Kevin Fall and Kannan Varadhan, editors. *The ns Manual*. The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, 14 April 2002. World Wide Web, http://www.isi.edu/nsnam/ns/doc/. Ongoing.

[5] Gregory R Ganger, Gregg Economou, and Stanley M Bielski. Self-securing network interfaces: What, why and how, Carnegie Mellon University Technical Report CMU-CS-02-144, August 2002.

[6] Jeffrey O Kephart and Steve R White. Directed-graph epidemiological models of computer viruses. In *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 343–359, May 1991.

[7] Jeffrey O Kephart and Steve R White. Measuring and modeling computer virus prevalence. In *Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 2–15, May 1993.

[8] A G McKendrick. Applications of mathematics to medical problems. In *Proceedings of Edin. Math. Society*, volume 14, pages 98–130, 1926.

[9] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. Brite: Universal topology generation from a user's perspective. Technical Report BUCS-TR2001-003, Boston University, 2001. World Wide Web, http://www.cs.bu.edu/brite/publications/.

[10] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver. Inside the slammer worm. In *IEEE Security and Privacy journal, 2003*, 2003.

[11] David Moore, Colleen Shannon, Geoffrey Voelker, and Stefan Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, April 2003.

[12] Sumeet Singh, Cristian Estan, George Varghese, and Stefan Savage. The earlybird system for real-time detection of unknown worms. Paper submitted to HOTNETS-II, August 2003.

[13] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the internet in your spare time. In *Proceedings of the* 11^{th} *USENIX Security Symposium*, August 2002.

[14] `CERT Advisory CA-`2003-04. Ms-sql server worm. World Wide Web, http://www.cert.org/advisories/CA-2001-19.html, 2003.

[15] Yang Wang, Deepayan Chakrabarti, Chenxi Wang, and Christos Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *Proceedings of the 22nd International Symposium on Reliable Distributed Systems*, 2003.

[16] Yang Wang and Chenxi Wang. Modeling the effects of timing parameters on virus propagation. In *Proceedings of the 2003 ACM workshop on Rapid Malcode*, pages 61–66. ACM Press, 2003.

[17] Matthew M Williamson. Throttling viruses: Restricting propagation to defeat malicious mobile code. Technical Report HPL-2002-172, HP Laboratories Bristol, 17 June 2002.

[18] Cliff Changchun Zou, Lixin Gao, Weibo Gong, and Don Towsley. Monitoring and early warning for internet worms. In *Proceedings of the 10th ACM conference on Computer and communication security*, 2003.

[19] Cliff Changchun Zou, Weibo Gong, and Don Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the* 9^{th} *ACM Conference on Computer and Communication Security*, November 2002.