Preliminary Results Using Scale-Down to Explore Worm Dynamics

Nicholas Weaver[*] ICSI

Ihab Hamadeh[†] Penn State

George Kesidis[‡] Penn State

Vern Paxson[§] ICSI

ABSTRACT

A major challenge when attempting to analyze and model large-scale Internet phenomena such as the dynamics of global worm propagation is finding appropriate abstractions that allow us to tractably grapple with size of the artifact while still capturing its most salient properties. We present initial results from investigating "scaledown" techniques for approximating global Internet worm dynamics by shrinking the effective size of the network under study. We explore scaledown in the context of both simulation and analysis, using as a calibration touchstone an attempt to reproduce the empirically observed behavior of the Slammer worm, which exhibited a peculiar decline in average per-worm scanning rate not seen in other worms (except for the later Witty worm, which exhibited similar propagation dynamics). We develop a series of abstract models approximating Slammer's Internet propagation and demonstrate that such modeling appears to require incorporating both heterogeneous clustering of infectibles and heterogeneous access-link bandwidths connecting those clusters to the Internet core. We demonstrate the viability of scaledown but also explore two important artifacts it introduces: heightened variability of results, and biasing the worm towards earlier propagation.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Invasive software

General Terms

Security

Keywords

worms, simulation, scaledown, modeling, Slammer

1. INTRODUCTION

A major challenge when attempting to analyze and model large-scale Internet phenomena such as the dynamics of global worm propagation is finding appropriate abstractions that allow us to tractably grapple with size of the artifact while still capturing its most salient properties. The general problem of faithful Internet-scale simulation, modeling, and analysis is well-recognized as exceedingly challenging [3]; but we still might hope to find certain problem domains which prove amenable to "scaledown" techniques by which we can reduce the effective size of the Internet as a means of making analysis tractable.

In this paper we explore scaledown techqniques for approximating the global worm dynamics exhibited by a particular class of worms, "bandwidth-limited scanners" [12]. We do so primarily in the context of the Slammer worm [6], which was the fastest Internet worm observed to date, spreading worldwide in less than 10 minutes. As we will discuss, Slammer exhibited a peculiar propagation dynamic: a steep reduction in the scanning-rate-per-worm, quite unlike the constant scanning rate exhibited by previous random-scanning worms such as Code Red[13, 8, 10]. This phenomenon appears directly related to network saturation, as additional copies of Slammer sharing the same access link are unable to contribute to the worm's aggregate scanning rate. We use this distinguishing feature as a touchstone to gauge the accuracy of the different abstract models we develop, and to then assess how differing degrees of scaledown affect that accuracy.

Unlike other forms of worms, the spread of bandwidth-limited scanning worms is limited by the available network bandwidth: each instance of the worm pumps out contagion as fast as it can, which often means that the worm's traffic can completely consume the available capacity along its path to the Internet core. To date, in addition to Slammer there has been only one other bandwidth-limited scanning worm of note, Witty [7, 11], which infected 12,000 systems in a span of 45 minutes. Below we use some measurements of Witty as way to partially double-check our Slammer analysis, although a complete analysis of Witty remains future work.

In Section 2, we detail the technology of scanning worms such as Code Red, as well as a discussion of bandwidth-limited scanning worms. This encompasses several subtypes of worms, of which Slammer and Witty represent the fastest and simplest subtype. We follow this with a discussion of Slammer's Internet spread in Section 3. By using empirical data for Slammer's scanning rate, combined with a mathematical model of the fraction of systems infected, we are able to calculate the average scanning-rate-per-worm. Although the scanning rate itself increases drastically over the course of the epidemic, the average scanning rate drops precipitously, apparently due to multiple copies sharing network links.

Section 4 then discusses the general problem of scaledown. For a complex worm model, we are unable to efficiently simulate the entire Internet address space. The solution is to reduce the address

[*] Email: nweaver@icsi.berkeley.edu

[†] Email: hamadeh@cse.psu.edu

[‡] Email: kesidis@engr.psu.edu

[§] Email: vern@icir.org

WORM04 October 29, 2004, Washington, DC, USA

space and vulnerable population by the same factor, but the process of scaledown adds two significant artifacts, a shifting zero point and additional noise, when compared to a full address space simulation of a conventional (Code-Red-like) scanning worm. Some of these artifacts are apparent in the mathematical model, while others only occur in simulation.

In Section 5 we introduce our first model of Slammer, the homogeneous cluster model. This model treats all worms as existing in identically sized clusters, which is simple enough to simulate with a modified version of a previous simulator. We discuss this simulator in Section 6. We show that the homogeneous cluster model captures the most significant feature of a decreased scanning-rate-per-worm, but deviates substantially from the measured reality. We also demonstrate how the same scaledown artifacts seen when scaling a Code-Red-like simulation also appear in this more complex model, as well as an additional scaledown artifact.

Section 7 introduces our second model for Slammer, the heterogeneous cluster model. This model uses known network information to cluster the worm instances, although all clusters have identical access links. Section 8 describes how we simulated the models, including execution on the DETER testbed [2]. Using a $1/64$ scaledown factor enabled us to achieve realtime emulation. This model provides better fidelity when compared to the homogeneous model, although scaledown-related artifacts become visible.

Finally, we summarize our future work in Section 9, and offer conclusions on scaledown, our model's fidelity and the effects of scaledown in Section 10.

2. SCANNING WORMS

Scanning worms [10, 4] operate by picking "random" addresses and attempting to infect them, for some variation of random (including biases for local addresses, randomly selected linear scanning, and randomly selected subblocks). The classic random scanning worm, Code Red [8], was relatively simple. Each worm had 100 threads, with each thread picking a random address to attempt to contact with a `connect()` system call. Since the connect often had a long timeout, the average scanning rate was roughly 6 scans per second per worm, regardless of the access link capacity, as Code Red used relatively little network bandwidth.

The spread of a conventional random scanning worm, such as Code Red, can be modeled as a random-constant-spread [10], where all copies have the same average scanning rate. This model can be generally expressed as the logistic function:

$$A = \frac{e^{K(T-t_0)}}{1+e^{K(T-t_0)}}$$

which represents both the fraction of the vulnerable population infected and the identical fraction of the aggregate scanning rate as a function of time. The spread rate parameter K is equal to the vulnerability density times the scanning rate, while t_0 is an offset which insures that at $T = 0$, only a single machine is infected.

Thus, since the spreading rate depends on K, the faster a worm can scan the more effective it can spread. For worms like Code Red, the scanning rate is generally limited by either network latency (the time it takes for the `connect()` call to receive a positive or negative response) or timeout (for when the `connect()` call receives no acknowledgment).

Bandwidth-limited scanning worms such as Slammer[1] are not limited by the latency of the infection attempts. Instead, they are limited by how fast the outbound link can be used to send traffic. Thus if several copies share the same access link, the first copy can often achieve the maximum scanning rate, with subsequent copies reducing the average scanning rate for the worm's group. Thus the basic assumption behind the random-constant-spread model doesn't hold: the average scanning rate per worm *changes* as the infection progresses.

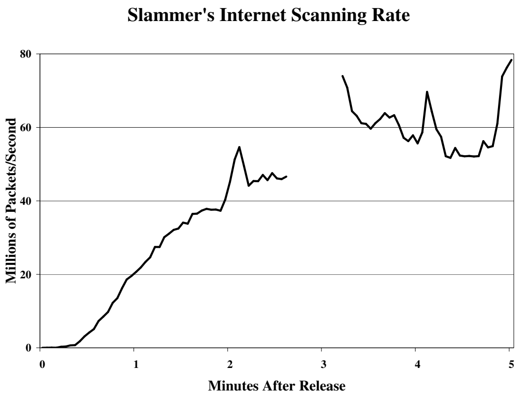

Figure 1: Slammer's Internet scanning rate, as measured at the University of Wisconsin Tarpit Network

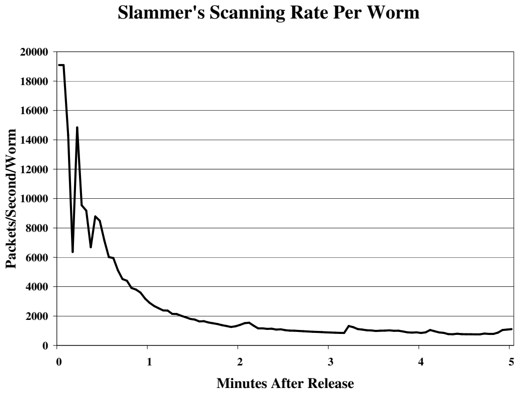

Figure 2: Slammer's scanning-rate-per-worm, as derived from the scans seen by the University of Wisconsin Tarpit Network

3. SLAMMER'S INTERNET SPREAD

Slammer spread very quickly, infecting almost all of the vulnerable population within 10 minutes of release [6]. Yet beyond just spreading fast, Slammer was the first significant worm without a constant scanning rate.

Figure 1 shows Slammer's Internet scanning rate, as measured at the University of Wisconsin "tarpit" network. Using the instantaneous scanning rate we can estimate the total number of scans seen by that point in time. We can then also estimate[2] the fraction of the 75,000 victims infects at that point based on the probability

[1] Although Slammer's pseudo-random number generator was significantly flawed [6], the aggregate scanning can still be considered as truely random because of how the generator is seeded.

[2] A question remains here whether some systems scan only briefly before becoming disconnected from the network (for example, because their scanning traffic crashes a switch or routers near their access point). While our available data for Slammer does not allow us to assess this possibility directly, indirect evidence suggests it

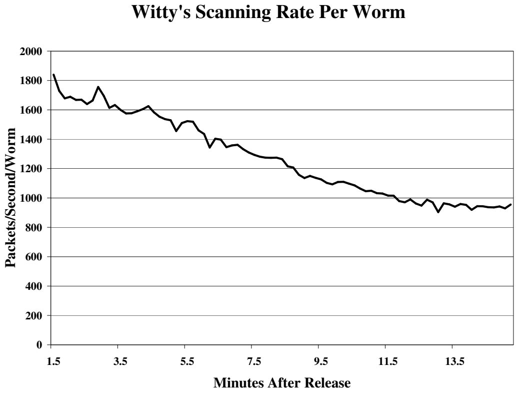

Figure 3: The Internet spread of witty: Witty's scanning-rate-per-worm, as measured by the CAIDA telescope for the time between 1.5 minutes and 15.5 minutes of Witty's propagation.

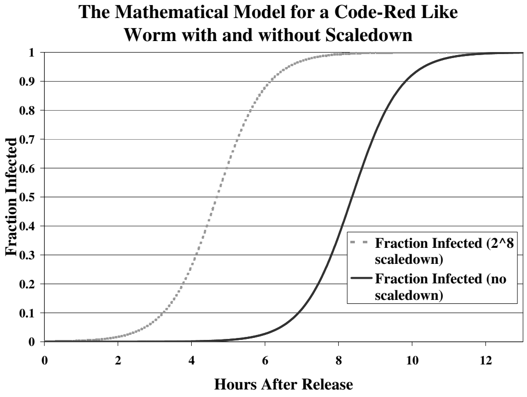

Figure 4: The effect of scaling down the mathematical model for Code Red. With a $1/2^8$ scaledown factor, the curve shifts substantially to the left because the *proportion* of the infected population at time t_0 is larger

of a system being infected after k scans, $P = 1 - (1 - 1/2^{32})^k$. Dividing this number into the scanning rate allows us to calculate scanning-rate-per-worm (Figure 2), which drops rapidly from an initial peak of over 19,000 scans-per-second-per-worm to slightly over 800 scans/s/worm over the course of the infection.

This represents a behavioral signature not seen in previous worms. However, it makes sense: an initial Slammer infection at a site would often saturate the outbound link, meaning that subsequent infections would not increase the aggregate scanning rate. Thus, although the total scanning rate continues to increase, the average scanning rate per worm drops precipitously during the course of the infection.

This behavioral signature can also be seen in the Witty worm, which is the only other single-packet UDP worm seen to date. Examining the incoming scans as observed at the CAIDA telescope, we can determine both the instantaneous scanning rate and the number of infections which are active at that point in time. We assume that an infection is active at a given point if a packet has been seen from that source both before and after that point in time.

The resulting scanning-rate-per-worm is shown in Figure 3. This does not display as clean a signature as that seen by Slammer, but this is a limitation of our dataset, as we don't have the first 90 seconds of Witty's propagation. Also, since Witty infected intrusion detection systems, we might expect (but cannot yet confirm) that fewer copies would share access links. Additionally, we deliberately undercount the number of active infections near the end of the trace, because our trace is only for 15 minutes. But even for this limited window, we can clearly see the reduction in scanning-rate-per-worm.

Thus, the general behavioral signature—a radically falling scanning-rate-per-worm—appears to be a generic signature of bandwidth-limited scanning worms rather than an artifact of Slammer in particular. As noted earlier, given this signature, we can then use the degree to which a proposed model matches the signature as a test of the model's fidelity.

4. SCALEDOWN

A basic scaledown technique is to take a large, full address space model and shrink it. Although doing so might not be necessary for simple models (i.e., we can simulate or analyze them directly at full scale), it becomes essential as models grow more complex. Yet by scaling down a simple and well understood model, we can determine what phenomena are introduced by scaledown.

For a worm experiment, we proceed by dividing the number of possible victims by a scaledown factor S. We similarly reduce the address space by the same factor, while keeping the scanning rate and other properties constant. For the mathematical model of a random scanning worm such as Code Red (Section 2), the value k will remain constant as it is dependent on the *density* of victims (not their number), which does not change due to scaledown.

But the overall results do change somewhat. We need to change t_0, the time offset, as a single infected machine (the starting point) represents a larger fraction of the population infected, shifting the curve to the left. Figure 4 shows this effect for a Code-Red-like worm. This is the first major scaledown artifact we observe.

Another scaledown effect concerns stochastic fluctuations (which we might term "luck"), which can play a significant factor in a worm's spread. As a simple example, consider the initial infection. If the initial infection discovers a new victim significantly earlier than expected, this can have a huge impact on the overall spread of the worm.

Figure 5 shows this in practice. We performed 10 simulation runs using the simulator initially developed for [10], with the same parameters and only different initial random seeds. The graph shows the mathematical model also displayed as a thicker line. Even for 300,000 possible victims in a 32 bit address space, luck plays a significant role in the time it takes a worm to spread.

These stochastic effects are further amplified when a simulation is scaled down to a smaller address space. When there are many infected systems, the probabilities average to a far greater degree than when there are a few systems. Thus when an experiment is scaled down, these effects become more significant, adding substantial noise to the propagation curves (Figure 6). Eventually, discrete effects dominate completely (Figure 7), as the smooth curves becomes very distorted over the entire range.

This can also be seen in Figure 8. This histogram shows the

is a minor effect, and our data for Witty *does* allow us to assess it directly, and confirms that for Witty it was minor.

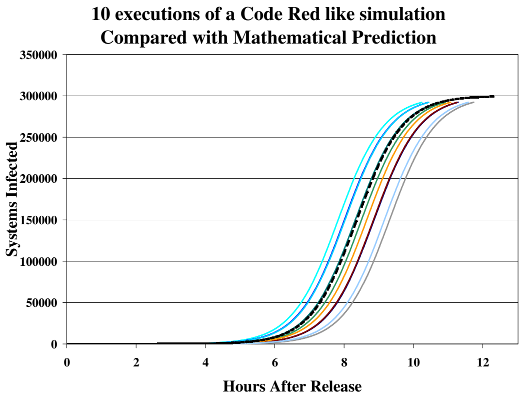

Figure 5: 10 simulation runs of a Code-Red-like worm, compared with the mathematical prediction (thicker, dashed line).

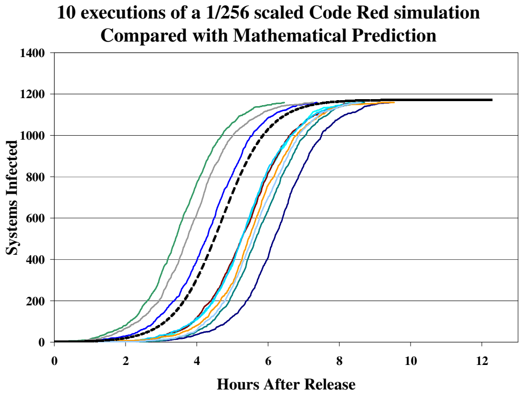

Figure 6: 10 simulation runs of a Code-Red-like worm with $1/2^8$ scaledown, compared with the mathematical prediction (thicker, dashed line).

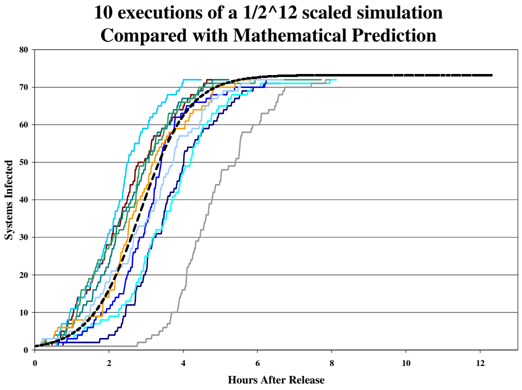

Figure 7: 10 simulation runs of a Code-Red-like worm with $1/2^{12}$ scaledown, compared with the mathematical prediction (thicker, dashed line).

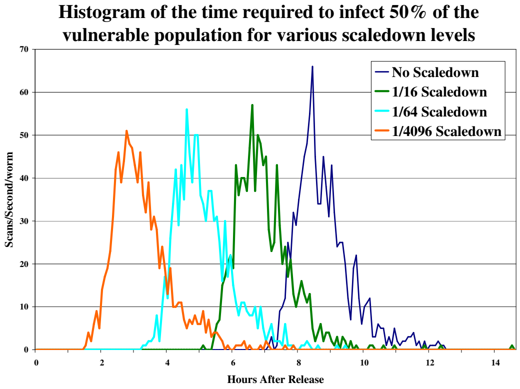

Figure 8: A histogram of 1,000 simulation runs for Code Red without scaledown, with a $1/2^4$, a $1/2^8$, and a $1/2^{12}$ scaledown factor. The histogram shows the time when 50% of the vulnerable systems are infected.

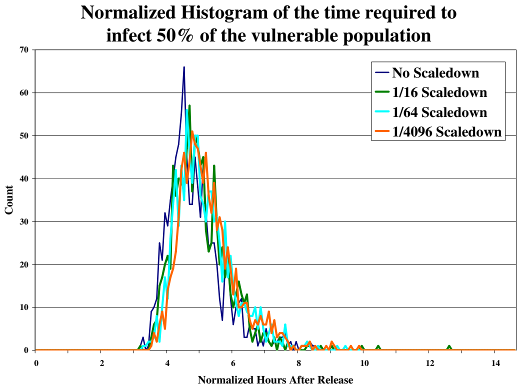

Figure 9: A histogram of 1,000 simulation runs for Code Red without scaledown, with a $1/2^4$, a $1/2^8$, and a $1/2^{12}$ scaledown factor. Unlike the previous histogram, here we have shifted the different histograms so that t_0 is equivalent.

results for 1,000 simulation runs for each of four scaledown factors (no scaledown, $1/2^4$ scaledown, $1/2^8$ scaledown, and $1/2^{12}$ scaledown). The histogram reports the time when 50% infection is achieved. As the scaledown factor is increased, the center of the histogram shifts to the left. This is the same phenomenon seen mathematically in Figure 4: the increasing scaledown shifts the curve to the left by changing t_0.

Additionally, as stochastic factors become more significant, the histogram for the higher levels of scaledown spreads out over a wider area. This effect is less significant, but can be seen in Figure 9. For this graph, we normalized the curves to the $1/2^8$ scaledown, to remove the effects of the shifting t_0. The higher scaledown factors spread out the curve more to the right, decreasing somewhat the absolute peak of the histogram. However, we do not yet have an explanation for the slight shift of the peak to the left.

Thus, although scaledown is effective and useful, it is critical to limit the amount of scaledown employed in practice, as the discrete effects can become substantial. Unlike the shifting of t_0, we cannot compensate for these discrete effects using a simple mathematical tweak.

5. HOMOGENEOUS CLUSTER MODEL

As we established in Section 3, Slammer's behavior does not match the assumptions of the Random Constant Spread model or other models of propagation. We observed that this deviation is probably due to bottleneck-bandwidth effects: multiple copies sharing the same access link will compete with each other, with the first infection able to start spreading at full rate, but with subsequent infections not offering any additional benefit.

We first attempt to model this in an abstract manner. We assume that the core of the Internet is "perfect": full bandwidth and zero latency. We can safely ignore latency because it has little effect on Slammer's spread, as Slammer did not need to receive acknowledgments. Additionally, worst-case latency is only in 100s of milliseconds, while Slammer's initial doubling time had an 8 second period [6]. Likewise, although many access links saturated, the backbone itself continued to operate during Slammer's spread.

We model each infectible system as existing in a cluster of C systems which all share a common access link to the core. It is this access link which represents the bottleneck, allowing only S scans/second to traverse onto the Internet. We argue that this is a reasonable assumption, as during Slammer's propagation access links tended to behave in one of three ways: crashing immediately, crashing after BGP sessions were dropped [5], or remaining up.

If the link crashed immediately, no system behind the access link would contribute to Slammer's Internet spread. Yet these systems would not be observed in our estimate of the infected population, allowing us to ignore this effect. Likewise, if a link only crashes after several minutes of propagating Slammer, this would have no effect on Slammer's propagation as most systems were compromised within the first 5 minutes.

We don't believe that a significant number of links crashed on a timescale of $>$5 minutes, as Slammer's aggregate scanning rate remained constant for three hours [6]. Likewise, when we examined our trace for Witty, over 1/2 the systems have a lifetime of > 5 minutes, despite Witty having a "kill the host" routine which drastically shortens system lifetime. Similarly, in our trace, only 20% of the systems are not seen in the last minute, suggesting that 80% of the systems maintained connectivity throughout the 15 minute trace of Witty's propagation.

Our first model uses homogeneous clusters: all clusters have the same link capacity and the same number of susceptible systems. As we will see in Section 6, this model captures the most significant behavior: the rapidly decreasing scanning-rate-per-worm curve, but fails to provide a detailed match.

6. SIMULATING THE HOMOGENEOUS MODEL

To investigate the homogeneous model using simulation, we modified the worm simulator used in [10] to include clustering as an option. For a parameter set, we used 73,728 systems in a 32 bit address space, with 18 systems per cluster and a cluster scanning rate of 15,000 scans/second. These parameters were chosen to approximately match the Slammer population and the beginning and ending scanning-rate-per-worm figures associated with Slammer.

Figure 10 shows the scanning-rate results of 10 simulations of the homogeneous cluster model, compared with the empirically

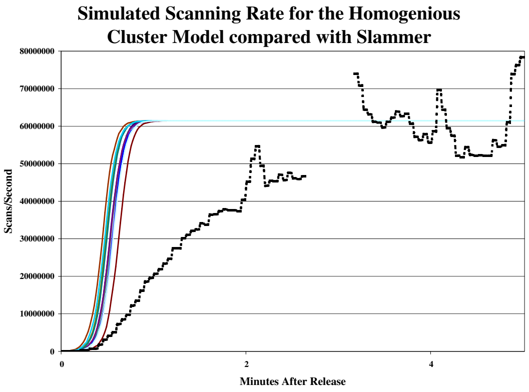

Figure 10: 10 simulation runs for the homogeneous cluster model, compared with empirically observed results (thicker, dashed line).

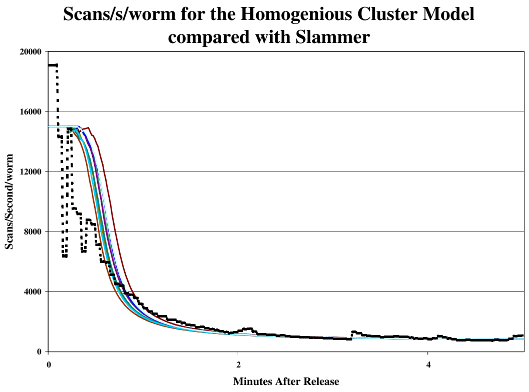

Figure 11: 10 simulation runs, showing the scanning rate per worm for the homogeneous cluster model, compared with empirically observed results (thicker, dashed line).

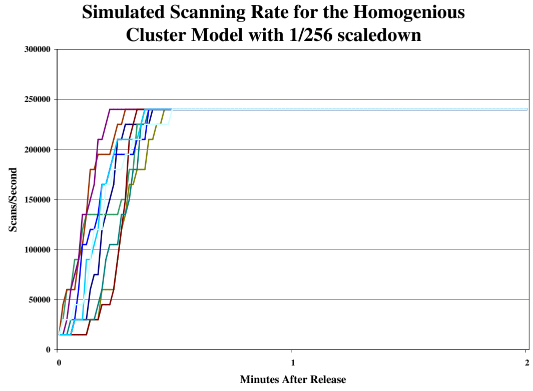

Figure 12: 10 simulation runs, showing the scanning rate for the homogeneous cluster model with a scaledown factor of $1/2^8$

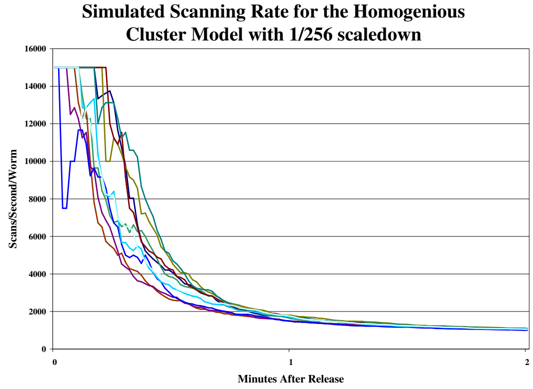

Figure 13: 10 simulation runs, showing the scanning rate per worm for the homogeneous cluster model with a $1/2^8$ scaledown factor.

measured values, while Figure 11 shows the scanning-rate-per-worm evolution. As can be seen, the homogeneous cluster model presents the most salient feature: the radically reduced scanning-rate-per-worm graph. Yet this deviates substantially in the details when compared to reality, suggesting an incomplete model.

However, this model is sufficient to observe the effects of scaledown. When we scaledown the model, not only do we see the shift to the left and the spreading of the scanning-rate curves (Figure 12), we also see a drastic increase in noise on the scanning-rate-per-worm curves (Figure 13).

This noise is also due to discrete effects. Although a small amount of noise appears in the system without scaledown, it greatly increases with increased scaledown, as the probability of infecting a new cluster versus an already existing cluster becomes more significant. Thus, although scaledown works effectively in this model, it does introduce an additional artifact when compared with scaledown for a Code-Red-like worm: significant noise in the scanning-rate-per-worm graph.

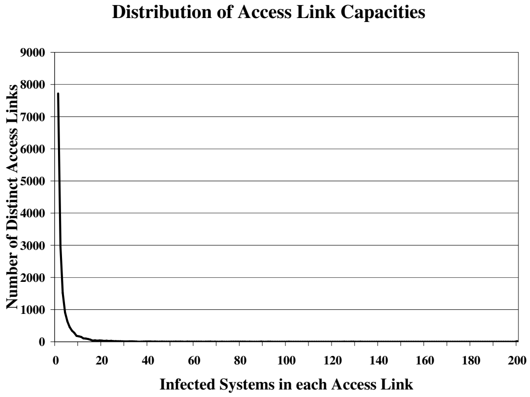

Figure 14: The distribution of the number of BGP prefixes by the number of infected systems in each BGP prefix

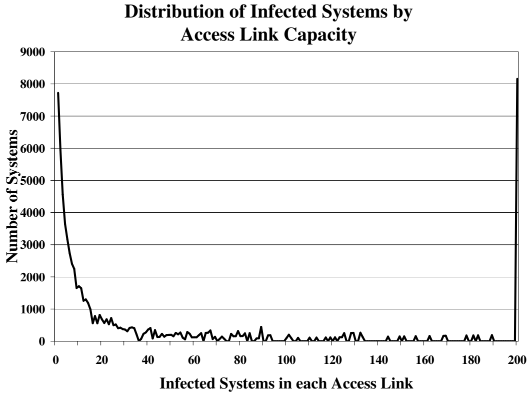

Figure 15: The distribution of the number of worms by the number of infected systems in each BGP prefix

7. HETEROGENEOUS CLUSTER MODEL

Since the homogeneous model is obviously insufficient, we attempt to refine our model using previous observations of Slammer's spread. We began by trying to obtain from providers topology maps of portions of the Internet which included bandwidth information, but this proved difficult, and while we did eventually acquire a significant amount of data in this regard, analyzing it and incorporating it into our modeling remains for future work. For the present, we instead simply approximate the access link distribution.

We used the BGP information from RouteViews [9] to get a list of all routed prefixes. Using this list, we mapped each Slammer infectee into the most precise prefix. We then operated under the assumption that each routed prefix uses an independent access link, so only infectees that shared a prefix might interfere with one another.

We graph the distribution of the access-links by the number of worms in Figure 14. The plot shows that most access links have only a few susceptible machines; that is, the typical access link did not service a large cluster of infectees.

The next graph, Figure 15, plots the distribution of the number of infectees by the cluster size the infectee resided in. It shows, unlike the first plot, that the typical infectee *did* find itself in a large cluster of infectees. The two figures together convey that there were a few

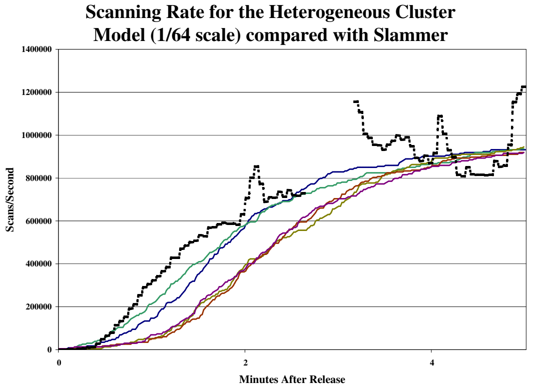

Figure 16: The scanning rate for 5 runs of the scaled heterogeneous model, compared with a $1/64$ scale of the empirical results for Slammer (thicker, dashed line)

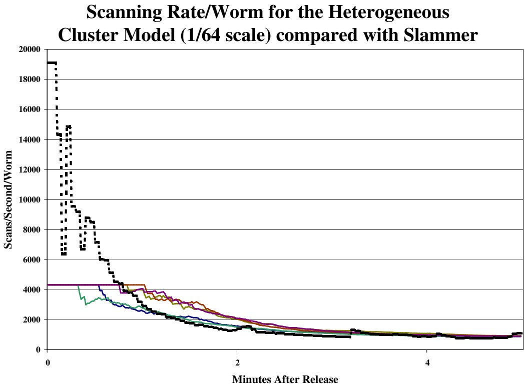

Figure 17: The scanning rate per worm for 5 runs of the scaled heterogeneous model, compared with empirical results for Slammer(thicker, dashed line)

quite-large clusters of infections; thus, most access links only had a few infections, but most infectees were part of a few large clusters.

We now can formulate a heterogeneous clustering model, in which we use the empirical distribution of clusters-per-prefix as shown in Figure 14, but retain an approximation that all of the access links have identical capacity. This gives us a model of the Internet where the susceptible machines are clustered based on assumed topology, but all the access links are identical.

8. SIMULATING THE HETEROGENEOUS CLUSTER MODEL

Once we have this model, we can then directly simulate it in a scaled-down form on the DETER [2] testbed. Our simulator uses a message passing mechanism to communicate worm-infection-attempts between systems which emulate large clusters of infectible systems. This simulation is able to emulate the worm in real-time, as a building block for testing larger defense systems.

To meet the realtime constraints, we needed to scale the experiment down by $1/64$. Since this is a more complex, realtime simulator running on an actual testbed system, we must use at least this scaledown factor in order to meet the constraints of the testbed. In order to generate the topology used in the experiment, we randomly selected access links, and added them to our simulation. We repeated this process until the total number of systems behind the selected access links was $1/64$th of the total number of infected systems observed. Doing so keeps the density of infected systems constant, and selects a distribution of access links comparable to the observed distribution. To set the access link bandwidth, we selected a value which matched the final scanning-rate-per-worm, which gives us a scanning-rate per access link of 4,300 scans/second.

As can be seen in Figure 16 and Figure 17, the heterogeneous cluster model produces a better set of results, more accurately matching the smoothness of Slammer's scanning rate and scanning-rate-per-worm curves. However, that the model is still not perfect suggests that the remaining homogeneous assumption of uniform access link capacity does not suffice. Thus, we have shown that we cannot assume a homogeneous distribution of infectible machines, and we cannot treat the access links as identical. If we wish to improve our model further, we will need to account for the varying capacities of the access links, not just the distribution of infected machines.

9. FUTURE WORK

We are currently striving to improve our model by accounting for access link capacities, since simply assuming all links are identical is insufficient. Thus, to increase the fidelity of our model, we will require a more refined Internet topology model.

Additionally, we are currently investigating Witty to verify that our modeling framework is generic for this class of worm. Although we have an initial indication, our datasets are currently not complete, and we have yet to fully analyze Witty's propagation in the same detail which we analyzed Slammer's.

We are also working on a Kermack-McKendrick model [1] that captures both the homogeneous and heterogeneous cluster models. Although a K-M model cannot capture the stochastic effects, it should be useful in verifying our simulations and modeling.

10. CONCLUSIONS

Scaledown is an important technique for modeling Internet-scale events, as it can make an otherwise intractable simulation tractable. For the case of a classic Code-Red-like scanning worm, scaledown introduces two notable artifacts: a bias towards more rapid propagation (the "shift to the left") and an increase in stochastic effects. Although these artifacts are significant, scaledown can still capture general behavior as long as the scaledown factor is not too extreme.

We have observed a particular behavioral signature for bandwidth-limited worms such as Slammer and Witty: a dropping scanning-rate-per-worm. We have developed two models, the homogeneous cluster model and the heterogeneous cluster model, which replicate this behavior, suggesting that access link congestion is the source of this signature.

The homogeneous model, although of lower fidelity, can be explored via a full simulation. Since we have a full simulator, we can verify the effects of scaledown for other experiments.

The heterogeneous cluster model uses AS topology (globally visible network prefixes) as an approximation for actual network topology. Our simulator runs in realtime, with tolerable ($1/64$) scaledown on the DETER testbed. This model is more accurate, but still insufficient, suggesting that the assumption of uniform access link capacity is insufficient to model Slammer. Instead, a high-fidelity model will require a more refined model of Internet topology.

11. ACKNOWLEDGMENTS

Our thanks to Abhishek Kumar for analyzing the scanning rate data for Witty, and Colleen Shannon and David Moore (CAIDA) and Vinod Yegneswaran and Paul Barford (WAIL) for providing traces of Slammer and Witty. Funding has been provided in part by NSF under grant ITR/ANI-0205519 and by NSF/DHS under grant NRT-0335290.

12. REFERENCES

[1] D. Daley and J. Gani. *Epidemic modeling, an introduction*. Cambridge University Press, 1999.

[2] Deter: Cyber defense technology experimental research (deter) network, http://www.isi.edu/deter/.

[3] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4), Aug. 2001.

[4] P. R. J. Mirkovic, J. Martin. A taxonomy of DDoS attacks and DDoS defense mechanisms.

[5] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang. An analysis of bgp update burst during slammer attack. In *Proceedings of the 5th International Workshop on Distributed Computing (IWDC)*, December 2003.

[6] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Magazine of Security and Privacy*, pages 33–39, July/August 2003 2003.

[7] D. Moore and C. Shannon. The Spread of the Witty Worm, http://www.caida.org/analysis/security/witty/.

[8] D. Moore, C. Shannon, and k claffy. Code-Red: a Case Study on the Spread and Victims of an Internet Worm. In *Proceedings of the Second Internet Measurement Workshop*, pages 273–284, November 2002.

[9] University of oregon route views project, http://www.routeviews.org/.

[10] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in your spare time. In *Proc. USENIX Security Symposium*, pages 149–167, Aug. 2002.

[11] N. Weaver and D. Ellis. Reflections on witty: Analyzing the attacker. *;login:*, pages 34–37, June 2004.

[12] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A Taxonomy of Computer Worms. In *The First ACM Workshop on Rapid Malcode (WORM)*, 2003.

[13] C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proc. ACM Conference on Computer and Communication Security*, 2002.