On the Race of Worms, Alerts and Patches

Milan Vojnović and Ayalvadi Ganesh

***Abstract*— We study the effectiveness of automatic patching and quantify the speed of patch or alert dissemination required for worm containment. We focus on random scanning as this is representative of current generation worms, though smarter strategies exist. We find that even such "dumb" worms require very fast patching. Our primary focus is on how delays due to worm detection and patch generation and dissemination affect worm spread. Motivated by scalability and trust issues, we consider a hierarchical system where network hosts are partitioned into subnets, each containing a patch server (termed superhost). Patches are disseminated to superhosts through an overlay connecting them and, after verification, to end hosts within subnets. When patch dissemination delay on the overlay is negligible, we find that the number of hosts infected is exponential in the ratio of worm infection rate to patch rate. This implies strong constraints on the time to disseminate, verify and install patches in order for it to be effective. We also provide bounds that account for alert or patch dissemination delay. Finally, we evaluate the use of filtering in combination with patching and show that it can substantially improve worm containment. The results accommodate a variety of overlays through the novel abstraction of a minimum broadcast curve. They demonstrate that automatic patching can be effective if combined with mechanisms to bound worm scan rate and with careful engineering of the patch dissemination. The results are obtained analytically and verified by simulations.**

***Index Terms*— Patching, Software Updates, Automatic Updates, Epidemic, Worm, Virus, Minimum Broadcast Curve**

I. Introduction

MOST Internet worms observed to date have been reverse engineered from patches released to address some vulnerability. The time between patch release and worm appearance has been shrinking; e.g., the Witty worm [5] was observed little more than a day after patch release. It may soon happen that a worm appears before the patch is available (a.k.a. *zero-day worms*). How can we limit the spread of such a worm? The doubling time of the number of infected hosts for some worms is in the order of tens of seconds, which means Internet scale infection is possible within minutes. Clearly, human response is too slow and a worm containment system *must be automatic*. An automatic patching system requires (i) worm detection, (ii) patch generation, and (iii) patch dissemination, verification and installation. Our work is primarily concerned with item (iii). The first two items have been considered in work by others, discussed later. Incorporating the effect of delays due to items (i) and (ii) through appropriate choice of the initial conditions (number of infected hosts), we consider the system from the time instant at which the worm has been detected and a patch generated. From then on, the two processes of patch dissemination from the server at which it was generated, and of worm spread from the population of infected hosts, are in a race. Our objective is to patch vulnerable hosts before they can be infected. How fast does patch dissemination have to be in order to win the race, i.e., to limit worm spread to a specified multiple of the number of initially infected hosts? Note that we do not propose a new automatic patching system, but rather provide analytical results that apply to a broad set of patching systems. We complement and verify our results by simulations.

Author Affiliations: M. Vojnović is with Microsoft Research, Cambridge (e-mail: milanv@microsoft.com). A. Ganesh is with Microsoft Research, Cambridge (e-mail: ajg@microsoft.com). A short version of this paper was presented at the workshop ACM-SIGSAC WORM'05, Fairfax, VA, USA, November 2005.

A. Motivation

If vulnerable hosts could be patched many orders of magnitude faster than the worm can spread, then the problem would be rendered of marginal interest. But this is not a safe assumption in practice. In fact, current practice relies on the assumption that the time between disclosure of a vulnerability and the appearance of a worm exploiting is sufficiently long, so that the majority of hosts can be patched before the onset of the worm. For example, with central server systems, such as the present-day Microsoft Windows Update, about 80% of distinct IP machines can be patched within a day, with the mean time until a query from a specific host equal to about 20 hours [4]. Comparing this with the timescale of some worms, such as the aforementioned Witty, it is clear that the system is not designed to contain zero-day or almost zero-day worms. This difficulty could be ameliorated by using a content distribution service [1] and by increasing the frequency of host queries, but this is unlikely to be a viable solution in practice due to the service provisioning costs involved. This would make the patch a faster runner in the race with a worm, but worms can also be made faster by using smarter strategies such as subnet-biased scanning, so the arms race between worms and countermeasures is likely to continue for some time into the future. Hence, a quantitative understanding of how the relative speeds of worm and patch influence the outcome of the epidemic is important.

B. Related Work

Much work has been done on studying worms and their containment [8], [11], [14]. We do not aim at addressing all related work, but only that which to our knowledge is closely related to automatic patching. There are several schemes for worm detection, e.g., (i) honeypots: these monitor unused segments of IP address space. The presumption is that scans to these addresses are either due to malicious attempts or misconfigured protocols (ii) detecting anomalous scanning

[1] By way of illustration, Windows Update serves about 300 million clients. If these were partitioned perfectly among Akamai's 15,000 servers, each server would need to serve patches to about 20,000 client machines.

behaviour either at end hosts or in the network (iii) detecting worm signatures either by looking for common patterns in network traffic or by analysing data and control flow of computer program executions. Automatic patch generation is addressed in Vigilante, which was proposed by Costa et al [3], and motivates the work in this paper. Vigilante is an end-host based system for automatic worm containment. Its main feature is that when a host detects a worm, it generates a self-certifying alert (SCA). Such alerts identify a vulnerability and provide a machine-generated proof of it which can be verified by any recipient. Vigilante uses a structured overlay to propagate alerts to all hosts in the system, which then use some mechanism to generate a filter which essentially corresponds to a patch. The self-certifying property of alerts is important as it solves the problem of trust and the concomitant one of attacks using the automatic response system. A similar system was also proposed by Sidiroglou and Keromytis [13]. There has also been work on analysing the competing processes of patching, filtering and worm spread [11], [14], [17]. These works typically consider a 'flat' network for both the worm and patch processes, whereas we study a hierarchical model motivated by considerations of scalability and trust.

C. Structure of the Paper

In Section II, we discuss our system assumptions and summarise our results. Section III-B shows our results for a patching system. We first provide a closed-form estimate for the number of ultimately infected hosts, for a patching system with all subnets initially alerted (Corollary 1), which can be interpreted as an approximation for systems with fast dissemination of alerts among superhosts. The effect of alert dissemination time is considered in Theorem 3. Section IV addresses push-based patch delivery. In Section V, we study the patching system complemented with filters that block all incoming and outgoing worm scans for alerted subnets. We present our simulation results in Section VI and then conclude in Section VII. Proofs of our results are provided in the appendices.

II. System model and Summary of Results

A. Worm Model

We consider worms that scan the IP address space uniformly at random. This is the method used by many, but not all, worms observed so far. Smarter strategies include biasing the scan in favour of the local subnet (routing worms), and exploiting the topology of some application-layer overlay (e.g. instant messenger buddy lists, ssh address lists etc.) While the techniques described here can be extended to routing worms, topological worms appear to be a much harder problem. Random scanning worms are characterised by the worm infection rate β, which is the rate at which per-host scans hit vulnerable IP addresses. For concreteness, $\beta = 0.00045$ per second for CodeRed[2] and $\beta = 0.117$ per second for Slammer [7]. In our discussions, we use the infection rate of Slammer as a recurring example, but the reader should bear in mind that this worm exhibits global spread behaviour that is somewhat different from that of a random scanning worm; see [6] for an analysis of the underlying causes.

[2]See www.caida.org/analysis/security/code-red.

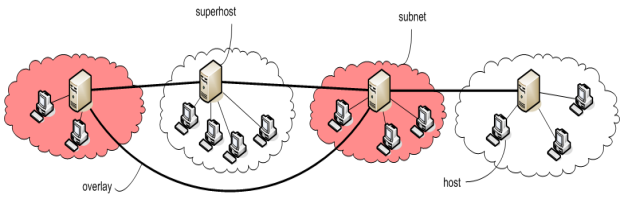

Fig. 1: Patching system illustration. Host address space is partitioned into subnets (e.g. ISP-like or IP prefix subnets). There is a superhost in each subnet, a dedicted patch sever. Superhosts are interconnected with an overlay. Each superhost (or subnet) is either alerted or dormant. An alerted superhost (i) provides patches to hosts in its own subnet and (ii) disseminates alerts to superhosts in other subnets. Hosts either pull the patch from their superhost or the patch is pushed from a superhost to a host. The highlighted subnets in the figure are alerted.

B. Patch Dissemination Model

1) Hosts, superhosts, subnets: The use of a centralised server to distribute patches to all clients is not scalable. Costa et al [3] consider a fully decentralised peer-to-peer scheme where hosts are organised into a structured overlay over which alerts/patches are spread. It is not clear whether such a system will be universally deployed. In this work, we consider an intermediate setting where hosts are partitioned into subnets and there is a patch server in each subnet, which we call a superhost. Superhosts are connected by an overlay, which is used to propagate patches between them. A superhost that has received a patch is said to be alerted, as is the subnet to which it belongs. Once a superhost is alerted, it propagates the patch to other superhosts as well as to hosts within its own subnet. We think of subnets as corresponding to network administration domains such as a university or corporation; one could also imagine ISPs maintaining a distributed set of superhosts. The superhost could be thought of as either enforcing a security policy by pushing patches on to clients, or as sending them self-certifying alerts (as in [3]), which the clients use to generate filters. The two are equivalent from a modelling perspective.

2) Alert dissemination to superhosts: Alerts are disseminated through the overlay connecting superhosts. The overlay can be arbitrary, e.g., a balanced multicast tree, one of several standard structured overlays (such as Pastry [10], Chord [15] or CAN [9]) or an unstructured overlay (such as Gnutella).

The impact of overlay topology on the performance of the patch dissemination system is captured through the new concept of a *minimum broadcast curve*, which is discussed in detail in Section III-E. Briefly, it abstracts the impact of overlay topology through a function which is a lower bound on the number of superhosts alerted over time. Such a lower bound can be computed easily for commonly used overlays.

3) Patch delivery within a subnet: Once a superhost is alerted, it has a patch for hosts within its own subnet and in addition propagates the alert to other superhosts. Patch dissemination within a subnet can be either by pull or push. In the former, patch delivery is initiated by a query from the host, while in the latter, it is initiated by superhosts. In the pull model, we assume that each host queries its superhost at the instants of a Poisson process of rate μ. In the push model, we assume that each superhost has an inventory of hosts in its own subnet. On becoming alerted, it pushes patches to them sequentially (in arbitrary order) at rate $N\mu$, where N is the number of hosts in the subnet. Thus, the mean load on the patch server is the same in both models. We show in Section IV that the push system performs somewhat better.

In the pull model, we assumed that inter-query times at each host are exponentially distributed with mean $1/\mu$. This is analytically convenient as it simplifies the dynamics due to the memoryless property of the exponential distribution. We show in Section III-E that this assumption entails no significant loss of generality; see the Remark following Propostion 2.

C. Summary of the Results

1) Required Frequency of Host Queries: Simple analysis shows that in the absence of countermeasures, the characteristic timescale of epidemic spread is $1/\beta$ (the time to double the number of infected hosts is $\log 2/\beta$) and most hosts are infected within some fairly small multiple of this time; for example, the time to go from 100 infected hosts to 1 million is about $10/\beta$. This suggests that any reactive countermeasure needs to have response time smaller than or comparable to the worm characteristic time. Note that the time $1/\beta$ is about 40 minutes for Code Red and smaller than 10 seconds for Slammer. In the case of a worm like Slammer, this suggests that we either need to patch most hosts within a timescale of about 1 minute or deploy additional mechanisms like rate capping [16] in order to slow down the worm.

In an idealised scenario, when alert dissemination between superhosts is instantaneous, we obtain an exact expression relating the initial and final number of infected hosts, for a given worm infection rate β and patching rate μ; see Corollary 1. The result implies that the ratio of final to initial infectives is at most $\exp((1-p(0))\beta/\mu)$, where $p(0)$ is the fraction of hosts that are initially patched (e.g. because the patching has started earlier than the worm appeared). Thus, for instance, presuming that no host is initially patched, the final number infected can be limited to no more than 100 times the initial by taking $\mu > \beta/5$. This is the sort of operating regime in which we are interested. In particular, we find that we need $1/\mu$ to be about 3 hours for Code Red and 50 seconds for Slammer. If a subnet contains 1000 hosts, this means that we need to patch at a rate of 5 hosts per minute for Code Red and 20 hosts per second for Slammer.

2) Minimum Broadcast Curve: More realistically, we are interested in the situation when alert dissemination to superhosts is not instantaneous. The time it takes to alert superhosts depends on the broadcast mechanism and the shape of the overlay used. We abstract these details by introducing the concept of a minimum broadcast curve: a function $a(t)$, $t \geq 0$, is said to be a minimum broadcast curve for an overlay of superhosts if, for any broadcast of alerts initiated at time 0, the fraction of alerted superhosts of the overlay at time t is at least $a(t)$. We can explicitly compute the minimum broadcast curve for simple topologies (see [1] for details). It is given by (i) exponential function (truncated at 1) for a tree, and (ii) logistic function for hypercubes and for gossip-based dissemination. We also verify that when flooding alerts on a Pastry [10] overlay, the fraction of alerted superhosts over time is well approximated by a logistic function.

We can use the minimum broadcast curve to obtain an upper bound on the fraction of hosts that eventually become infected; the results are presented in Theorem 2, for the specific case of a logistic minimum broadcast curve. Theorem 3 extends the results to the case when μ and β are both small, i.e., alert dissemination is fast compared to both worm spread and patching. Simulation results presented later show that this formula is indeed accurate in realistic parameter regimes.

3) Patching and Filtering: We investigate the effectiveness of patching combined with filters that block worm scans in and out of alerted subnets, and obtain a closed-form solution for the fraction of infected and susceptible hosts at any time within non-alerted subnets. The eventual fraction of infected hosts within a subnet is now entirely determined by the fraction of infected hosts in the subnet at the time instant when it became alerted.

III. Patching

A. System Assumptions

In this section we consider a hierarchical patch distribution where, in the first layer, the patch is disseminated among superhosts and in the second layer, superhosts make the patch available to hosts within their subnet. We have a population of N hosts stratified into J disjoint subnets, with the jth subnet being of size N_j. Associated with each subnet is a superhost, denoted by the same index j. A host is either susceptible, infected or patched. We assume that patched hosts cannot be infected by the worm and, conservatively, that once infected, a host cannot be patched (on the timescale of our automatic patching system; this subsumes the case where the 'patch' is actually an alert that enables filtering out attacks, but not curing the infection). An infected host attempts to infect other hosts by scanning the address space of size Ω uniformly at random at a fixed rate η. We say that a superhost is alerted if it has received a patch and dormant otherwise; in a system like Vigilante, we could say that a superhost is alerted when it has received and verified an alert, and generated a filter for that vulnerability.

Dormant superhosts do nothing. Active superhosts make the patch available for hosts within their subnet to pull; each host in the subnet becomes patched after a random time which is exponentially distributed with mean $1/\mu$. In addition, active superhosts disseminate alerts among themselves. We shall make some specific assumptions on this dissemination process and show how the framework accommodates a variety of alert dissemination systems. We analyse these models by

considering an asymptotic regime (large population limit) in which Ω, N and J increase to infinity, while the ratios Ω/N and J/N are kept fixed.

Time scale. Without loss of generality, we scale time by κ, the rate of alert dissemination, and henceforth take the rate of alert dissemination to be unity. All our statements can be interpreted without difficulty by merely replacing the rates β and μ and times t with β/κ, μ/κ and κt respectively.

B. Host Population Dynamics

Denote by $i(t)$ and $s(t)$ the fraction of hosts which are infected and susceptible respectively, at time t. Similarly, let $i_A(t)$ and $s_A(t)$ be the fraction of all hosts which are infected and susceptible respectively, and reside under alerted superhosts. Denote by $a(t)$ the fraction of superhosts which are alerted. We assume that superhosts cannot be infected by the worm, e.g., because they do not run services exhibiting the vulnerability. The system dynamics is as follows:

$$\begin{aligned} a'(t) &= a(t)(1-a(t)) && (1)\\ i'(t) &= \beta i(t)s(t) && (2)\\ s'(t) &= -\beta i(t)s(t) - \mu s_A(t) && (3)\\ i'_A(t) &= \beta i(t)s_A(t) + a(t)(i(t)-i_A(t)) && (4)\\ s'_A(t) &= -(\beta i(t)+\mu)s_A(t) + a(t)(s(t)-s_A(t)) && (5) \end{aligned}$$

This system of differential equations describes host population dynamics and is justified in the limit of many hosts and many superhosts in the precise sense described in Appendix A.

The last term in (4) comes from a subnet whose superhost undergoes transition from dormant to alerted state. This happens at rate $Ja(t)(1-a(t))$ and, when it does, the number of infectives belonging to subnets with alerted superhosts jumps up by the number of infectives in that subnet, which is $N[i(t)-i_A(t)]/[J(1-a(t))]$ on average. The last term in the derivative for $s_A(t)$ is explained similarly; see Appendix A for details.

We can solve for $a(t)$ explicitly:

$$a(t) = \frac{a(0)}{a(0)+(1-a(0))e^{-t}}. \qquad (6)$$

Any alert dissemination mechanism that results in the same time evolution of $a(t)$ as in (6), namely a logistic function, will yield the same solution for the dynamics (1)–(5). In particular, the results apply equally to alert dissemination by gossip or over a hypercube, or commonly used structured overlays such as Pastry.

We recast the system of equations (1)–(5) into an equivalent form by defining the auxiliary process $w(t) = s_A(t)/s(t)$. In words, $w(t)$ is the fraction of all susceptible hosts at time t which belong to subnets that have been alerted by this time. We have

$$\begin{aligned} \frac{d}{dt}i(t) &= \beta i(t)s(t), && (7)\\ \frac{d}{dt}s(t) &= -\beta i(t)s(t) - \mu\, w(t)s(t), && (8)\\ \frac{d}{dt}w(t) &= \mu w(t)^2 - (\mu + a(t))\, w(t) + a(t). && (9) \end{aligned}$$

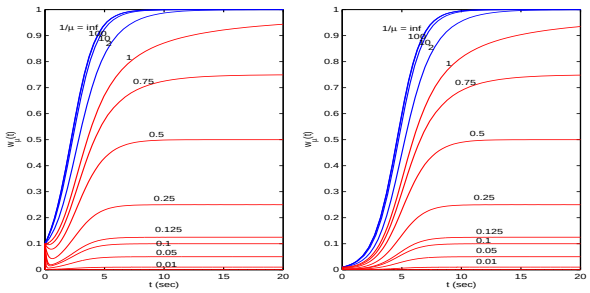

Fig. 2: Fraction of susceptible hosts that belong to alerted subnets, $w(t)$, for $w(0) = a(0) = 1/10$ (left) and $w(0) = a(0) = 1/100$ (right). Different curves correspond to different values of $1/\mu$, in seconds. Convergence to limit points 1 and $1/\mu$ is well exhibited.

Equations (7)–(8) are similar to those one would obtain for single-subnet patching, but with a time-dependent hazard function, $\mu w(t)$, of host queries; the extra $w(t)$ term corresponds to the fraction of hosts that have access to patches at time t. The differential equation (9) is known as Ricatti's equation and can be solved in closed form.

Proposition 1. *The solution for w is given by*

$$w(t) = 1 - \frac{e^{\mu t}}{1-a(0)+a(0)e^{t}}\frac{1}{\Phi_\mu(t) - z(0)}, \qquad (10)$$

where $z(0) = -1/[1-w(0)]$, and

$$\Phi_\mu(t) = \int_1^{e^{\mu t}} \frac{dx}{1-a(0)+a(0)x^{1/\mu}}. \qquad (11)$$

Moreover, the limit point of $w(t)$ is specified as follows:

$$\lim_{t\to+\infty} w(t) = \begin{cases} 1, & \text{if } \mu \le 1,\\ \frac{1}{\mu} & \text{if } \mu > 1. \end{cases}$$

Interpretation. As t tends to ∞, both the number of susceptible hosts $s(t)$ and the number of such hosts belonging to subnets with an active superhost, $s_A(t)$, tend to zero at an exponential rate. Hence, in looking at the limit of $w(t)$ as $t \to \infty$, we are asking about the relative speeds and the manner in which $s(t)$ and $s_A(t)$ tend to zero.

If $\mu \le 1$, then the per host patching rate is slower than the spread of the epidemic process describing the activation of superhosts. Hence, new superhosts get alerted faster than hosts within the subnet are patched; eventually, all superhosts have been alerted but only a small fraction of hosts have been patched. From this time onwards, the system behaves like a single large unstratified network. Thus, in this case, most susceptibles (a fraction approaching 1) belong to a subnet with an active superhost; it is the per-host patching rate μ which is the bottleneck.

On the other hand, if $\mu > 1$, then patching within a subnet is faster than the activation of new superhosts. A non-negligible fraction of hosts within the subnet have been patched before the superhost activates a new superhost. By the time most superhosts have been activated, most hosts in active

subnets have also been patched, and a substantial fraction of susceptible hosts belongs to those few subnets where the superhost is still dormant. In this case, the system behaves like a collection of subnets that are activated sequentially rather than like one big subnet where patching is active from the start. Here, it is the rate of alert dissemination among superhosts that limits the speed with which patching is achieved.

Comment. A somewhat similar host immunisation process to (7)–(8) was considered by Wong et al [17]. They do not consider patch dissemination on an overlay, hence $w(t) \equiv 1$. The major distinction is that Wong et al (Section 6.1 [17]) assume a host can be patched in both infected and susceptible state. This amounts to the following system:

$$\begin{aligned} \frac{d}{dt}i(t) &= \beta i(t)s(t) - \mu i(t), \\ \frac{d}{dt}s(t) &= -\beta i(t)s(t) - \mu s(t). \end{aligned}$$

In contrast, we assume that an infected host cannot be patched over the timescale of patch dissemination, which could be the case, e.g., for smart worms which disable patching at an infected host. Also, as noted earlier, this is a realistic assumption for systems which disseminate alerts rather than patches; here, the alert can be used to generate filters which guard against infection, but cannot cure an existing infection.

C. Ultimately Infected Hosts

We now present a main theorem that allows derivation of more explicit results later:

Theorem 1. *For the system of differential equations (1)–(5), it holds that*

$$i(+\infty) + \frac{\mu}{\beta}\int_0^{+\infty} w(u)d\log i(u) = 1 - p(0). \tag{12}$$

where, recall, $p(0)$ is the fraction of initially patched hosts.

The following corollary is of interest. It provides an implicit function for the number of ultimately infected hosts, for the special case with all superhosts initially alerted.

Corollary 1. *Assume $a(0) = 1$, i.e. all superhosts are initially alerted. Then, $w(\cdot) \equiv 1$ and it follows that*

$$i(+\infty) + \frac{\mu}{\beta}\log\frac{i(+\infty)}{i(0)} = 1 - p(0). \tag{13}$$

Indeed, in this special case, the network behaves like a single large subnet. The result may be regarded as an approximation for the limiting case in which patch dissemination on the overlay connecting superhosts is much faster than either worm spread or patch spread within subnets.

Comment. Corollary 1 has the following implication:

$$i(+\infty) \le i(0)\exp\left((1-p(0))\frac{\beta}{\mu}\right). \tag{14}$$

In words, the number of hosts which ever become infected is a multiple of the number initially infected; this multiple is at most exponential in the ratio of worm infection rate β to patching rate μ. We see in Section VI that this bound is a good approximation for ranges of the value of β/μ that are of practical interest.

For further intuition on this bound, note that the fraction of infected hosts $i(t)$ for a random scanning worm with infection rate β, and with no patching, satisfies the inequality $i(t) \le i(0)\exp(\beta t)$, for any $t \ge 0$. Suppose now that the automatic patching system ensures that all vulnerable hosts are patched no later than time T, if not already infected by the worm. Then it follows from the above that $i(+\infty) \le \exp(\beta T)$. Now (14) is of the same form, but with the deterministic upper bound T replaced by the mean patching time $1/\mu$.

D. The Effect of Alert Broadcast Time

In Corollary 1, we consider the case with all superhosts initially alerted, which would hold by observing the system evolution from a time instant when all superhosts became alerted, and would be a good approximation for systems with fast dissemination of alerts. This yields a lower bound on the fraction of eventually infected hosts in a practical system with a non-zero alert time. Suppose now that $T > 0$ is a deterministic bound on the time to alert all superhosts; we call it an alert broadcast time. An upper bound on the fraction of hosts eventually infected can be obtained by making the worst-case assumption that no superhost is alerted prior to time T. Thus, we get:

Theorem 2. *Suppose the patching rate is μ in each subnet, and that the alert dissemination system guarantees that the alert broadcast time is at most T. Then, the fraction of ultimately infected hosts $i(+\infty)$ satisfies:*

$$\begin{aligned} & i(+\infty) + \frac{\mu}{\beta}\log\left(\frac{i(+\infty)}{i(0)}\frac{i(0)+s(0)e^{-\beta(1-p(0))T}}{i(0)+s(0)}\right) \\ \le\ & 1 - p(0). \end{aligned}$$

The inequality of the theorem implies

$$i(+\infty) + \frac{\mu}{\beta}\log\frac{i(+\infty)}{i(0)} \le (1-p(0))(1+\mu T).$$

As in the comment after Corollary 1, this implies

$$i(+\infty) \le i(0)\exp\left((1-p(0))\beta\left(\frac{1}{\mu}+T\right)\right). \tag{15}$$

Comparing the last inequality with (14), which holds when all superhosts are alerted at the start, we note that the alert broadcast time enters by effectively increasing the mean patching time from $1/\mu$ to $(1/\mu)+T$. This is a simple and intuitive result.

The inequality in the statement of Theorem 2 becomes tight as the worm infection rate β and patching rate μ both tend to zero. This limit regime corresponds to a separation of timescales whereby alert dissemination runs on a fast timescale compared to patching and worm spread. This regime is of practical interest. We formalise these statements in the following theorem, which applies to alert dissemination mechanisms characterised by a logistic function $a(t)$.

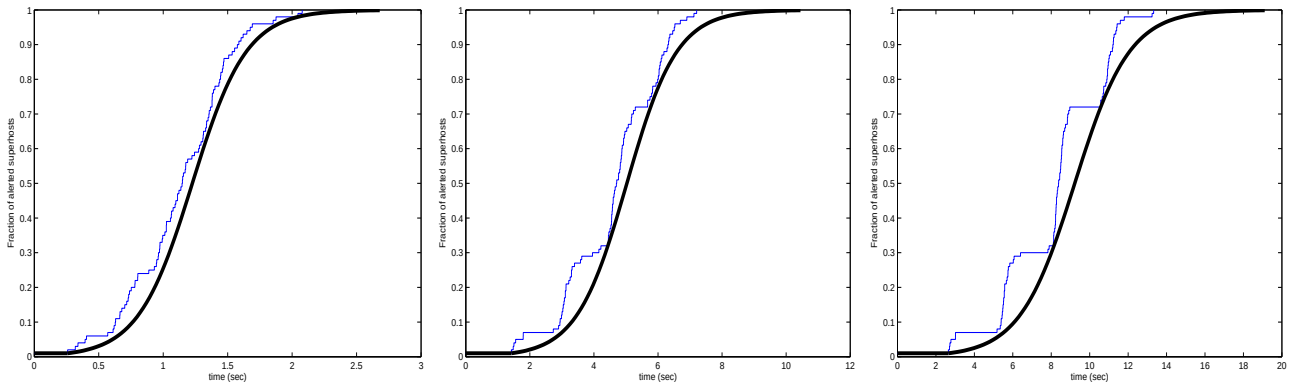

Fig. 3: Empirical fraction of alerted superhosts by flooding on a Pastry overlay of 100 superhosts and delayed-logistic minimum broadcast curves. (Left) the per-superhost delay = 0 seconds, (Middle) 1.16 seconds, and (Right) 2.38 seconds. The figure suggests delayed-logistic curve a natural candidate for a minimum broadcast curve of standard overlays.

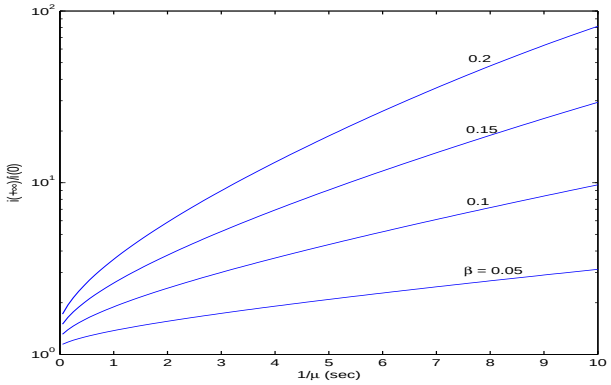

Fig. 4: The ultimate number of infected hosts with worm-like patch dissemination (22)–(24) versus mean patch time $1/\mu$. The initial values are $i(0) = 1/1000$, $a(0) = 1/10000$, $p(0) = 0$. The alert dissemination rate is $\kappa = 4$ sec^{-1}.

Theorem 3. *The patching system described by (1)–(5) satisfies*

$$\begin{aligned} & i(+\infty) + \frac{\mu}{\beta} \log\left(\frac{i(+\infty)}{i(0)}\right) \\ \sim \quad & 1 - p(0) + \mu \frac{s(0)(1-w(0))}{1-a(0)} \log\left(\frac{1}{a(0)}\right), \end{aligned}$$

where we write $\sim$ to mean that the ratio of the two sides tends to 1 as μ and β tend to zero with their ratio fixed.

Interpretation. The following heuristic argument provides intuition for the result. Suppose that initially, one subnet is alerted, so that $a(0) = 1/J$. Multiply both sides of the result of the theorem with β/μ and note that the right hand side scales as $\beta\left((1-p(0))(1/\mu) + K\log(J)\right)$, where K is appropriately defined constant. The alert broadcast time by random gossipping is indeed of the order $\log(J)$, so that the result simply amounts to enlarging the mean patch dissemination time with the mean alert broadcast time.

E. Minimum Broadcast Curve

The concept of *minimum broadcast curve* is an abstraction of the mechanism of alert broadcast on arbitrary overlay topologies. It is attractive as it allows us to abstract diverse broadcast networks by a single curve. We say that a function $m(t)$ is a minimum broadcast curve for a given alert dissemination system, if, defining $t = 0$ as the time of first alert, the fraction of alerted hosts on any interval $[0, t]$ is at least $m(t)$. Now note that the patching system (7)–(9) holds more generally than for $a(t)$ a logistic function (corresponding to alert dissemination by random gossip). It is intuitive to expect that if we replace $w(t)$ in (7) and (8) with some function that is a lower bound for all t, then the resulting solution yields an upper bound on the fraction of infected hosts for all t. This is the content of:

Proposition 2. *Consider functions f_1 and f_2 satisfying $0 \le m \le f_1(t), f_2(t) \le M < +\infty$, and the following two systems of differential equations:*

$$\begin{array}{rcll} \frac{d}{dt} i_j(t) & = & \beta i_j(t) s_j(t) & \\ \frac{d}{dt} s_j(t) & = & -\beta s_j(t) i_j(t) - f_j(t) s_j(t) & j = 1, 2. \end{array}$$

Assume that $f_1(t) \ge f_2(t)$ for all $t \ge 0$, and that on any finite interval $[s, t]$, there exists a finite partition $s \le t_1 \le \cdots \le t_{n-1} \le t_n = t$, such that (C):

$$\inf_{u \in [t_k, t_{k+1}]} f_1(u) \ge \sup_{u \in [t_k, t_{k+1}]} f_2(u), \text{ all } k = 1, 2, \ldots, n-1.$$

Then, $i_1(t) \le i_2(t)$ and $s_1(t) \ge s_2(t)$, for all $t \ge 0$, whenever $i_1(0) \le i_2(0)$ and $s_1(0) \ge s_2(0)$.

The conditions of the proposition are satisfied if the functions f_1 and f_2 are continuous, and $f_1(t) > f_2(t)$ for all t. Its relevance is that the solution of (7)–(9), using any minimum broadcast curve $m(t)$ in place of $a(t)$, yields an upper bound on the fraction of infected hosts in the actual system. In Figure 3, we compare the empirical broadcast curve obtained by flooding in Pastry and a minimum broadcast curve taken to be a logistic function; see [1] for details.

Remark: General query time distributions. We have assumed so far that inter-query times at each host are exponentially distributed with mean $1/\mu$. This is analytically convenient (it simplifies the dynamics due to the memoryless property of the exponential distribution) but is not essential.

Indeed, let $F(t)$ be an arbitrary cumulative distribution (CDF) of per-host inter-query times. The patch becomes available (superhost becomes alerted) at some arbitrary time, and we want to know how long it takes until a given host pulls the patch. By the renewal theorem, the residual time until the next pull event has the CDF:

$$F_r(t) = \mu \int_0^t (1 - F(x))dx,$$

where $1/\mu = \int_0^\infty (1 - F(x))dx$ is the mean time between pull attempts. From this, we can calculate the *hazard* function $\lambda(t)$, denoting the intensity that a query from a specific host is made at time t, given that the host made no query on the time interval $[0, t)$. The hazard function is given by

$$\lambda(t) = \frac{\frac{d}{dt}F_r(t)}{1 - F_r(t)}. \tag{16}$$

In the special case of exponentially distributed inter-query times, $F(t) = 1 - e^{-\mu t}$, and it can be readily verified that $F_r(t) = 1 - e^{-\mu t}$ and that $\lambda(t) \equiv \mu$, i.e., the hazard rate is constant. Now, for arbitrary hazard functions $\lambda(t)$, if $\lambda(t) > \lambda$ for all $t \geq 0$ and some $\lambda > 0$, then the pull system with inter-query time having CDF $F(\cdot)$ performs at least as well as the pull system with $Exp(\lambda)$ inter-query times. More precisely, the number of hosts that ever become infected is smaller in the former system. This result follows from the Proposition 2 above, and shows that there is no significant loss of generality in restricting attention to the exponential distribution.

Example (Windows Update) Consider the query times of a host to Windows Update service. Suppose the host is always on. Then the host query times form a renewal process with inter-query times uniformly distributed on $[a, b]$, where $(a, b) = (18, 22)$ hours [4]. Note that the host query rate is $\mu = 2/(a + b)$, thus 1 query per 20 hours. The residual time distribution of a host query is thus:

$$F_r(t) = \begin{cases} \mu t & 0 \leq t \leq a \\ 1 - \frac{(b-t)^2}{b^2 - a^2} & a < t \leq b. \end{cases}$$

It follows from (16) that

$$\lambda(t) = \begin{cases} \frac{\mu}{1-\mu t} & 0 \leq t \leq a \\ \frac{2}{b-t} & a < t < b, \end{cases}$$

and $\lambda(t)$ is undefined for $t \geq b$. It is clear that $\lambda(t) \geq \mu$ for all $t \in [0, b]$. Thus, by Propostion 2 above, the number of eventually infected hosts using the Windows Update patching system is no higher than if each host pulled patches at the instants of a Poisson process of rate once per 20 hours.

IV. Push-based Patch Dissemination

We have so far discussed a pull mechanism for patch dissemination, motivated by currently deployed systems. We now explore push schemes for comparison. We consider two approaches: a hierarchical scheme analogous to that studied for pull, and a peer-to-peer scheme, either system-wide or deployed within each subnet.

In the hierarchical scheme, each alerted superhost pushes patches on to the N nodes in its subnet in some order, at rate $N\mu$. Thus, the server capacity is taken to be the same as in the pull system considered earlier. For a single subnet, the system dynamics is:

$$\frac{d}{dt}i(t) = \beta i(t)s(t) \tag{17}$$

$$\frac{d}{dt}s(t) = -\beta s(t)i(t) - \mu\frac{1}{1-\mu t}s(t) \tag{18}$$

for $0 \leq t < 1/\mu$. Comparing with system (7)–(8), with $w(t) \equiv 1$, we note that (17)–(18) differs by the additional term $1/(1 - \mu t)$. Intuitively, for the same initial fractions of infectives and susceptibles and identical patch rate μ, the push system will result in a smaller number of infectives than the pull system. This intuition is indeed true in view of Proposition 2 above. The numerical values in Figure 5 illustrate the superiority of push, for the same server load.

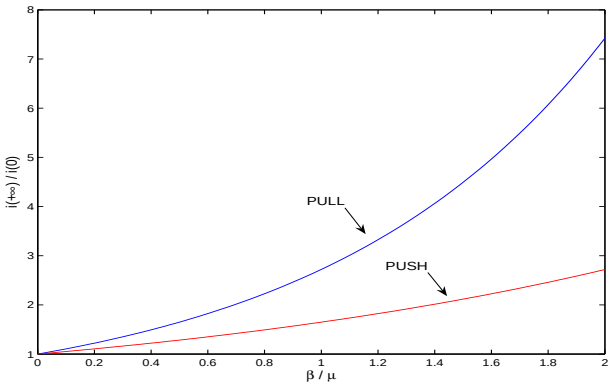

Fig. 5: The ratio of the ultimately infected hosts to initially infected hosts in a single subnet with pull and push patch dissemination. The initial fraction of infectives is 10^{-5}.

A. Worm-like Patch Dissemination

We noted above that a superhost can improve the effectiveness of patching by pushing patches to hosts rather than waiting for them to be pulled. Even better performance is achievable using peer-to-peer dissemination. Specifically, we consider gossip or epidemic-style protocols to disseminate the patch within subnets since such a scheme is fast, scalable, and robust. (The process of alert dissemination to superhosts remains the same as before.) We make the reasonable assumption that hosts have some side knowledge about addresses of other hosts in the same subnet and don't rely on random scanning to locate them. Then, the gossip scheme used for patch dissemination can proceed much faster than worm infection. This is modelled by taking the patch scan rate to be larger than the worm scan rate.

We distinguish two cases. In the first, we assume that as soon as a host receives a patch (or alert), it is quarantined and cannot be infected by the worm. This quarantine state lasts until the host verifies and installs the patch. In the second case, the host continues to be vulnerable in the interim. In both cases, we consider a single subnet. Equivalently, the analysis would apply if, instead of employing a hierarchical scheme,

the peer-to-peer scheme were deployed system-wide, as in Vigilante.

1) With Perfect Quarantine: We deal with two epidemics both spreading in the same manner, but the worm has smaller infection rate than the patch. Denote the worm infection rate by β and the patch infection rate by μ. Let $p(t)$ denote the fraction of patched hosts at time t. The race of worm and patch is specified by the following differential equations describing two competing epidemics:

$$\frac{d}{dt}i(t) = \beta i(t)(1 - i(t) - p(t)) \quad (19)$$

$$\frac{d}{dt}p(t) = \mu p(t)(1 - i(t) - p(t)). \quad (20)$$

The limit point of the system (19)–(20) follows readily.

Proposition 3. *The fraction of hosts ultimately infected, $i(+\infty)$, is the solution of the following equation*

$$i(+\infty) = i(0)\left(\frac{1 - i(+\infty)}{p(0)}\right)^{\frac{\beta}{\mu}}.$$

The proposition implies the following simple bound on the final fraction of infected hosts:

$$i(+\infty) \leq i(0)e^{\frac{\beta}{\mu}\log\left(\frac{1}{p(0)}\right)}. \quad (21)$$

The last inequality fleshes out the appeal of worm-like patch dissemination as μ can typically be chosen much larger than β. Then β/μ is close to 0 and, if the fraction of initially patched hosts $p(0)$ is not inordinately small, then so is $(\beta/\mu)\log(1/p(0))$. In other words, the fraction of hosts eventually infected exhibits only a small increase over the fraction initially infected at the time of worm detection.

Comparing (21) with (14), we see that it is (14), corresponding to the pull system, which yields the tighter bound on $i(+\infty)$, for *given* β and μ. But it should not be concluded from this that peer-to-peer patch dissemination performs worse than a centralised system! The explanation lies in the fact that while the patching load per host is proportional to μ in the peer-to-peer setting, the patching load per superhost scales like $N\mu$. So what the comparison really tells us is that peer-to-peer patch dissemination can achieve performance comparable to the hierarchical scheme with much smaller per-host load.

2) With Imperfect Quarantine: Suppose now that after a host receives a patch or alert, it is not patched instantaneously. Instead, there is some non-zero time required to verify the patch and install it. Such an assumption is required if trust cannot be assumed. During the time between receiving and installing a patch, the host continues to be vulnerable and will become infected if scanned by the worm. This may be a reasonable assumption if the worm can bypass quarantine. To facilitate further analysis, we assume that the time between receiving a patch and installing it is an exponentially distributed random variable with mean $1/\mu$ at each host, and independent across hosts. Denote by $a(t)$ the fraction of hosts that are alerted but not yet patched. We now have three epidemics in place that specify the race of worm, alert, and patch:

$$\frac{d}{dt}i(t) = \beta i(t)(1 - i(t) - p(t)) \quad (22)$$

$$\frac{d}{dt}a(t) = \kappa p(t)(1 - i(t) - p(t) - a(t)) - \beta i(t)a(t) - \mu a(t) \quad (23)$$

$$\frac{d}{dt}p(t) = \mu a(t). \quad (24)$$

Note that we assume that a host participates in dissemination of alerts only after it is patched; this is reflected in the factor $p(t)$ in (23). The reason for this assumption is that the time between alerting and patching includes the time for verifying the alert; the assumption is needed to prevent attacks based on enticing hosts to flood the network with bogus alerts.

The system of differential equations (22)–(24) can be solved numerically. Figure 4 shows the ultimate number of infected hosts for a range of values of β and $1/\mu$.The figure demonstrates that patch verification and installation time does significantly affect the number of ultimately infected hosts.

V. Filtering and Patching

We have studied a model with an overlay network of superhosts that acted as distribution servers for patches. In this section, we extend the model by letting superhosts also act as filters (or firewalls). Whenever a superhost j is alerted, a worm scan originating from or destined to the subnet of superhost j fails with probability 1. In other words, a worm scan at time t from subnet k to subnet m succeeds only if neither superhost k nor superhost m are alerted. As earlier, each host under an alerted superhost installs a patch with rate μ.

The race of worm and patch can be described by a Markov process (see [1]). Here we directly proceed to the limit population dynamics under the many hosts and many superhosts assumptions. We first consider the subpopulation of hosts in non-alerted subnets. Denote by $i_D(t)$ and $s_D(t)$ the fraction of infected and susceptible hosts in dormant (non-alerted) subnets. We have the following dynamics:

$$\frac{d}{dt}i_D(t) = \beta i_D(t)s_D(t) - a(t)i_D(t) \quad (25)$$

$$\frac{d}{dt}s_D(t) = -\beta s_D(t)i_D(t) - a(t)s_D(t) \quad (26)$$

$$\frac{d}{dt}a(t) = a(t)(1 - a(t)). \quad (27)$$

The equation for $a(t)$ is the same and has the same solution as given by (6). We obtain closed form solutions for i_D and s_D in this case.

Theorem 4. *The system of equations (25)–(26) has the solution:*

$$\begin{aligned} i_D(t) &= i_D(0)\frac{u(t)^{\beta'}}{\rho + (1-\rho)u(t)^{\beta'}}\frac{1 - a(0)u(t)}{1 - a(0)} \\ s_D(t) &= (1 - i_D(u(t)))\frac{1 - a(0)u(t)}{1 - a(0)}, \end{aligned} \quad (28)$$

with $u(t) := a(t)/a(0)$, $\beta' := \beta(i_D(0) + s_D(0))/(1 - a(0))$, and $\rho := s_D(0)/(i_D(0) + s_D(0))$.

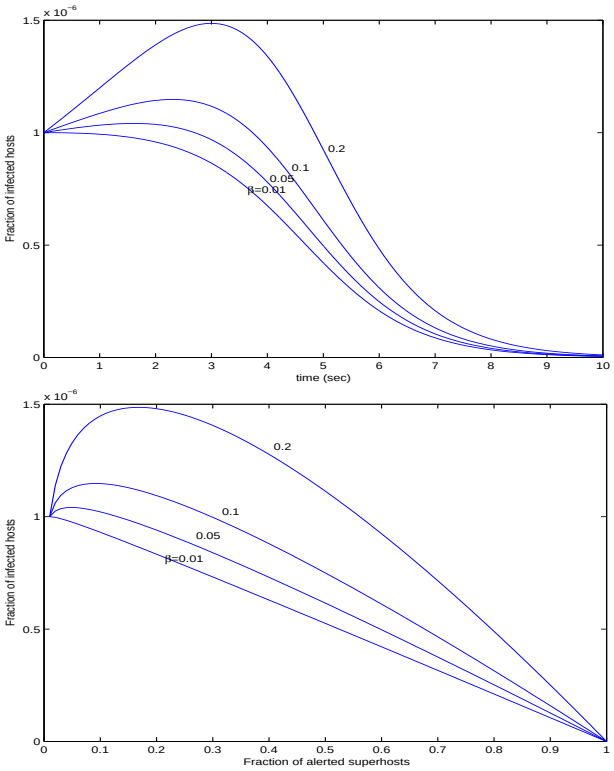

Fig. 6: (Top) The fraction of infected hosts in non-alerted subnets versus time, as given by (28). (Bottom) Same but versus the fraction of alerted superhosts.

The fraction of infected hosts $i_D(t)$ increases with t on some initial interval $[1, t_0)$, attains its maximum at some $t_0 > 0$ and then decreases to 0 as t goes to $+\infty$. The decrease is due to the fact that superhosts become alerted and so the fraction of non-alerted subnets decreases over time. See Figure 6 for numerical plots of the function $i_D(t)$. This is also validated in Section VI.

The reason for being interested in the evolution of $i_D(t)$ and $s_D(t)$ is that they can be used to obtain the number of infectives and susceptibles in a subnet j, at the time T_j at which superhost j becomes alerted. After this time, subnet j is isolated and its evolution decouples from that of the rest of the network. On $[0, T_j]$, we have

$$\frac{d}{dt} i_j(t) = \beta i_D(t) s_j(t), \quad (29)$$

$$\frac{d}{dt} s_j(t) = -\beta i_D(t) s_j(t). \quad (30)$$

Defining $\phi(t) := \exp\left(-\beta \int_0^t i_D(u) du\right)$, it follows that, in particular, $s_j(T_j) = s_j(0)\phi(T_j)$ and

$$i_j(T_j) = (1 - \phi(T_j)) n_j + \phi(T_j) i_j(0). \quad (31)$$

The function $\phi(t)$ can be obtained in closed-form by integrating the solution (28):

$$\phi(t) = \frac{i_D(0) + s_D(0)}{s_D(0) + i_D(0) u(t)^{\beta'}}.$$

Now, for $t > T_j$, we have

$$\frac{d}{dt} i_j(t) = \beta i_j(t) s_j(t)$$
$$\frac{d}{dt} s_j(t) = -\beta s_j(t) i_j(t) - \mu s_j(t).$$

But this is a familiar dynamics that we already encountered in Section III-B. We have an explicit relation between the initial and limit point of this system given in Corollary 1. Specialising to the current context, the identity of Corollary 1, for a subnet j, reads

$$i_j(+\infty) + \frac{\mu}{\beta} \log \frac{i_j(+\infty)}{i_j(T_j)} = n_j. \quad (32)$$

In view of the last identity, the ultimate fraction of infectives in a subnet is fully specified given $i_j(T_j)$, i.e. given the function $\phi(t)$ and the alert times T_j (see Equation (31)).

A. The Effectiveness of Filtering

We discuss filtering in the limit when alert propagation time is negligible, so that we can assume that $T_j = 0$ for all j. The benefit of filtering can be discerned from (32). To see this, suppose that n_j is of the order of $1/J$ (or exactly equal to $1/J$, assuming equal-sized subnets); then (32) implies $i_j(+\infty) \leq i_j(0) \exp((\beta/J)/\mu)$. Recall that this inequality is a good estimate for $i_j(+\infty)$ in the limit case where the first element in the left-hand side of (32) is much smaller than the second element. Summing over j, we have $i(+\infty) \leq i(0) \exp((\beta/J)/\mu)$, which is precisely the inequality that one would obtain for a patching system with worm infection rate β/J and patching rate μ. Thus, for the same initial conditions, the final number of infectives in a system with patching and filtering and worm infection rate β is bounded by the final number of infectives in the patching only system with worm infection rate β/J. In other words, filtering effectively reduces the speed of worm spread by a factor J equal to the number of subnets, which would be large in practice. This demonstrates the potential effectiveness of filtering in reducing worm infection rate.

We end this section by noting that, as in patching-only systems (Section III-B), we can use the abstraction of a minimum broadcast curve in patching and filtering systems to obtain an upper bound on the number of infected hosts. This is shown in [1, Proposition 3].

VI. Simulation Validation

Setup. We verify some of our results by packet-level discrete-event simulations in SimPastry [2]. In our simulations, superhosts are nodes in a Pastry overlay [10]. The Pastry nodes are attached to J nodes chosen uniformly at random from a network topology that is input to the simulator. We used a transit-stub topology generated by the Georgia Tech topology generator [18]. The topology has slightly more than 5000 routers arranged into two hierarchical levels: (top) 10 transit domains with approximately 5 routers in each transit domain; (bottom) 10 stub domains attached to each transit router with approximately 10 routers in each stub domain. The delays between routers are provided by the topology generator.

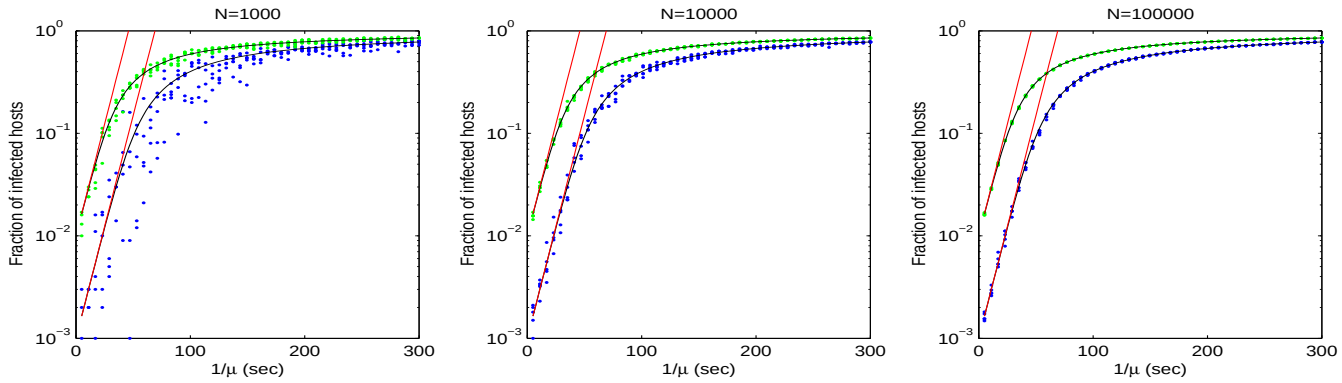

Fig. 7: Patching with all-alerted, equal-size subnets with the total number of hosts: (left) $N = 1000$, (middle) $N = 10000$, and (right) $N = 100000$. Each graph shows simulation results of two settings: (lighter) $i(0) = 1/100$ and (darker) $i(0) = 1/1000$. For each of the setting, there are 5 simulation outcomes obtained with distinct random seed. The worm infection rate is $\beta = 0.1$. The dark solid curves are that obtained by Corollary 1. The light solid lines are from (14).

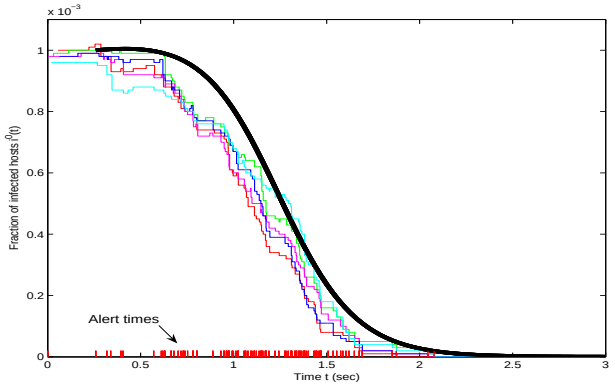

Fig. 8: Patching with filters and a Pastry overlay: Fraction of infected hosts in non-alerted subnets versus time. The worm scan rate is $\beta = 0.1$ and the initial fraction of infected hosts $i(0) = 1/1000$. The number of superhosts, J, is equal to 100. There are $N = 100000$ with 1000 hosts in each subnet. The graph shows sample paths of 5 simulation runs with random seeds. The thick lines are from Theorem 5, obtained for $a(t)$, a delayed-logistic minimum broadcast curve. The simulation results and analytical predictions match well.

The Pastry parameters are set as $b = 1$ and $\ell = 32$. The alert broadcast system is *flooding* performed as follows. Suppose a Pastry node becomes alerted at some time instant. It then broadcasts an alert to all other nodes that are in its routing table. Subsequently, each Pastry node receiving an alert, if not already alerted, broadcasts the alert to all the nodes in its routing table; otherwise, it discards the alert. This process continues until all Pastry nodes are alerted. We do not consider faults in this work. The alert broadcast time depends on the number of routing hops between any two nodes (diameter), and on the delays between nodes. In the presence of a worm, it may well happen that the network becomes saturated due to worm traffic and some servers slow down. To capture this, we vary as input parameter a fixed delay at each routing hop in the Pastry overlay. This lets us examine the performance under overload without making specific assumptions on the workload causing the delays. We implemented a packet-level worm propagation model. The model is that of a birth-and-death Markov process for infected and susceptible hosts found in [1] (Section III and Section V therein). The difference is that in the simulator, the alerts disseminate by flooding on Pastry. We next present our simulation results:

Patching with instantly alerted superhosts. The goal is to validate the result of Corollary 1 and the bound (14) on the ultimate fraction of infected hosts. We separate the effects of the alert broadcasting delays by configuring zero-valued delays at each link of the input network topology. See Figure 7 for the results; a detailed description of the simulation parameters can be found in the caption of the figure. The simulation results conform well to the analytical estimate of Corollary 1. Indeed, as expected, the larger the number of hosts N, the smaller the variability in the results. The figure also shows that the bound (14) is a good estimate as long as the number of ultimately infected hosts is sufficiently small. All the observations confirm predictions of the analysis.

Patching with alert delays. We also examined how our analysis predicts the fraction of ultimately infected hosts when alerts are propagated on overlay with some delay. We varied alert delays by adding a fixed per-hop delay in Pastry routing, ranging from 0 to 3 sec. The simulations results are in good agreement with the analytical prediction of Theorem 3; see [1].

Patching with filters. We consider the same simulation setup as with patching described earlier, but now, in addition, we have filters in place as described in Section V. We first validate Theorem 4. See the caption of Figure 8 for a description of the simulation setup. The figure provides a comparison of analytical and simulation results, and shows that there is a good agreement.

We also performed simulations to validate the identity (32). To that end, we used the same setup as in Figure 8, but varied a fixed per-hop-delay of alerts at each superhost. The results are not shown due to lack of space, but they confirm that: (i) patching with filters significantly outperforms a system with no filters; (ii) alert broadcast time is a significant factor.

VII. Conclusion & Discussion

Our analysis suggests that (i) patching can be effective only if the product of the fraction of the host population that is not patched and the ratio of the worm infection rate to patching rate is not too large. This would require the use of mechanisms such as per-host capping of scan rates in order to limit worm infection rates; (ii) for networks partitioned into subnets, with each subnet having a dedicated server for patch distribution (superhost), and superhosts connected by an overlay, the alert broadcast time on the overlay is a significant factor in determining the number of ultimately infected hosts; it is thus important to employ overlays with small diameter to ensure fast dissemination; (iii) the concept of minimum broadcast curve may prove useful as a unifying abstraction to capture alert dissemination delay over variety of overlays; (iv) worm-like dissemination of patches is an effective solution for patch dissemination, but only if the time to verify and install patches is not too large, presuming no host quarantining.

There are several problems to investigate further: (i) the implications of system heterogeneity with respect to subnet sizes and patching rates over subnets; (ii) extension of the framework to routing or local preference worms; (iii) the models provide a framework to study adversarial strategies for worms to maximise the ultimate number of infected hosts.

Acknowledgements

The work described here was inspired by conversations with Don Towsley. We thank Manuel Costa for his comments and simulator code, and Miguel Castro and Stuart Staniford for their valuable comments on an earlier draft of this paper.

References

[1] M. Vojnović and A. Ganesh. On the Race of Worms, Alerts, and Patches. Technical Report TR-2005-13, Microsoft Research, February 2005.

[2] M. Castro, P. Druschel, M. Jones, A.-M. Kermarrec, A. Rowstron, and M. Theimer. SimPastry version 1.1. http://www.research.microsoft.com /∼antr/pastry/download.htm, 2002.

[3] M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham. Vigilante: End-to-End Containment of Internet Worms. In *Proc. SOSP 2005*, Brighton, United Kingdom, October 2005.

[4] C. Gkantsidis, T. Karagiannis, P. Rodriguez, and M. Vojnović. Planet Scale Software Updates. In *Proc. ACM Sigcomm 2006*, Pisa, Italy, 2006.

[5] http://www.caida.org/analysis/security/witty, 2005.

[6] G. Kesidis, I. Hamadeh, and S. Jiwasurat. Coupled Kermack-McKendrick Model for Randomly Scanning Worms and Bandwidth-staturating Internet Worms. In *Proc. QoS-IP*, Sicily, Italy, February 2005.

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

[8] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proc. IEEE Infocom 2003*, San Francisco, CA, March 2003.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *ACM Sigcomm 2001*, 2001.

[10] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM Int'l Conf. on Distributed Systems Platforms (Middleware)*, pages 329–350, Heildelberg, Germany, November 2001.

[11] N. Weaver S. Staniford, V. Paxson. How to Own Internet in your Spare Time. In *IEEE Security & Privacy*, 2004.

[12] Adam Shwartz and Alam Weiss. *Large Deviations for Performance Analysis*. Chapman and Hall, London, 1995.

[13] S. Sidiroglou and A. D. Keromytis. Countering network worms through automatic patch generation. In *IEEE Security & Privacy*, 2005.

[14] S. Staniford. Containment of scanning worms in enterprise networks. *Journal of Computer Security (To Appear)*, 2004.

[15] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrisnan. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. on Networking*, 11(1), February 2003.

[16] M. M. Williamson. Throttling viruses: Restricting propagation to defeat malicious mobile code. In *ACSAC*, 2002.

[17] C. Wong, C. Wang, D. Song, S. Bielski, and G. R. Ganger. Dynamic quarantine of internet worms. In *Proc. of the International Conference on Dependable Systems and Networks (DSN-2004)*, Florence, Italy, June 2004.

[18] E. Zegura and S. Bhattacharjee. How to model an internetwork. In *Proc. of the INFOCOM'96*, San Francisco, California, 1996.

Appendix

A. Patching System (1)–(5), the Limit of Many Hosts and Many Superhosts

Consider a patching system with N vulnerable hosts partitioned into J subnets, with N_j vulnerable hosts in a subnet j. Let $\mathcal{J} = \{1, \ldots, J\}$. The patching system is described by the stochastic process $X(t)$ defined by:

$$\mathbf{X}(t) := (I_j(t), S_j(t), V_j(t), j \in \mathcal{J}),\ t \geq 0,$$

where $I_j(t)$ is the number of infected hosts in a subnet j, $S_j(t)$ is the number of susceptible hosts in a subnet j, and $V_j(t) = 1$ if and only if a subnet j is alerted, otherwise $V_j(t) = 0$. Here t is time, $t \geq 0$. Note that this is a complete state description as the number of patched hosts, $R_j(t)$, can be recovered from the identity $I_j(t) + S_j(t) + R_j(t) = N_j$, for all t. Denote with $I(t)$ and $S(t)$ the total number of infected and susceptible hosts at time t.

The stochastic process $\mathbf{X}(t)$ is Markov, with transition rates from a state $(\mathbf{I}, \mathbf{S}, \mathbf{V})$ to other states given as follows:

$$\begin{array}{ll} (\mathbf{I} + e_j, \mathbf{S}, \mathbf{V}) & \text{with rate } \frac{\eta}{\Omega}\left(\sum_{k=1}^{J} I_k\right) \cdot S_j \\ (\mathbf{I}, \mathbf{S} - e_j, \mathbf{V}) & \text{with rate } \frac{\eta}{\Omega}\left(\sum_{k=1}^{J} I_k\right) \cdot S_j + \mu V_j S_j \\ (\mathbf{I}, \mathbf{S}, \mathbf{V} + e_j) & \text{with rate } \frac{1}{J}\left(\sum_{k=1}^{J} V_k\right)(1 - V_j) \end{array} \tag{33}$$

where e_j is the J-dimensional vector with all elements equal to 0 except for the j^{th} element, which is equal to 1. Define

$$I_A(t) = \sum_{j=1}^{J} I_j(t) V_j(t), \quad S_A(t) = \sum_{j=1}^{J} S_j(t) V_j(t), \tag{34}$$

to be the number of infected and susceptible hosts belonging to subnets where the superhost is active at time t. Denote with $a^{(J)}(t) = \frac{1}{J}\sum_{j=1}^{J} V_j(t)$, the fraction of alerted superhosts at time t. From the transition rates of X, it follows:

$$\begin{aligned} I \to I + 1 &: \frac{\eta}{\Omega} IS \\ S \to S - 1 &: \frac{\eta}{\Omega} IS + \mu S_A \\ I_A \to I_A + 1 &: \frac{\eta}{\Omega} IS_A + a^{(J)}(I - I_A) \\ S_A \to S_A - 1 &: \frac{\eta}{\Omega} IS_A + \mu S_A + a^{(J)}(S - S_A) \\ a^{(J)} \to a^{(J)} + \frac{1}{J} &: J a^{(J)}(1 - a^{(J)}) \end{aligned}$$

Consider now a family of a sequence of stochastic processes, defined by scaling the space with parameter

N, $i^{(N)}(t) = I^{(N)}(t)/N$, $i_A^{(N)}(t) = I_A^{(N)}(t)/N$ and similarly for $s^{(N)}(t)$ and $s_A^{(N)}(t)$. We assume that $\beta = \eta N/\Omega$ is fixed as N tends to infinity. Note that the transition rates are functions only of the scaled state space, except the multiplier, either N or J, so the process is a density-dependent Markov jump process. We can thus invoke Kurtz's theorem [12] to assert that presuming $\lim_{N,J\to+\infty}(i^{(N)}(0), s^{(N)}(0), i_A^{(N)}(0), s_A^{(N)}(0), a^{(J)}(0)) = (i(0), s(0), i_A(0), s_A(0), a(0))$, where the latter if fixed, the Markov process $(i^{(N)}(t), s^{(N)}(t), i_A^{(N)}(t), s_A^{(N)}(t), a^{(J)}(t))$ uniformly converges on all compact intervals to the solution of (1)–(5) as N and J go to infinity.

B. Proof of Proposition 1

The differential equation (9) is known as Ricatti's differential equation. Noting that $v(t) \equiv 1$ is a particular solution of (9), we use the standard transformation $w_\mu(t) = v(t) + \frac{1}{z(t)} = 1 + \frac{1}{z(t)}$, to obtain the linear differential equation:

$$\frac{d}{dt}z(t) = (a(t) - \mu)z(t) - \mu.$$

This equation has the solution

$$z(t) = e^{-\mu t + \int_0^t a(u)du}\left(z(0) - \mu\int_0^t e^{\mu u - \int_0^u a(y)dy}du\right), \tag{35}$$

with $z(0) = \frac{-1}{1-w(0)} = \frac{s(0)}{s(0)-s_A(0)}$, as can be verified by differentiating $z(t)$. Now, we have from (6), after elementary calculations, that

$$\int_0^t a(u)du = \log\Big(1 - a(0) + a(0)e^t\Big). \tag{36}$$

Defining $\Phi_\mu(t) = \mu\int_0^t e^{\mu u - \int_0^u a(y)dy}du$, we see from (36) that

$$\Phi_\mu(t) = \mu\int_0^t \frac{e^{\mu u}}{1 - a(0) + a(0)e^u}du,$$

which yields (11) on making the change of variables $x = e^{\mu u}$. We also see from (35) and (36) that

$$w_\mu(t) = 1 - \frac{e^{\mu t}}{1 - a(0) + a(0)e^t}\frac{1}{\Phi_\mu(t) - z(0)},$$

which verifies (10).

It only remains to verify that $\lim_{t\to\infty} w(t)$ satisfies the claims of the proposition. We consider three cases.

1) $\mu < 1$: Observe from (11) that the integral in the definition of Φ_μ converges as $t \to \infty$. Hence, it is immediate from (10) that $\lim_{t\to\infty} w(t) = 1$.
2) $\mu = 1$: Observe that $\Phi_\mu(t) = \log(1 - a(0) + a(0)e^{\mu t}) \sim \mu t$, where we write $f(t) \sim a(t)$ to denote that $f(t)/a(t) \to 1$ as $t \to \infty$. Hence, by (10), $\lim_{t\to\infty} w(t) = 1$.
3) $\mu > 1$: We have by (11) that

$$\Phi_\mu(t) \sim \frac{1}{1 - (1/\mu)}\frac{x^{1-(1/\mu)}}{a(0)}\Big|_1^{e^{\mu t}} \sim \frac{\mu}{(\mu-1)a(0)}e^{(\mu-1)t}.$$

Substituting this in (10) yields $\lim_{t\to\infty} w(t) = 1/\mu$.

This completes the proof of the proposition.

C. Proof of Proposition 2

The proof uses an auxiliary lemma that is showed first:

Lemma 1. *Let $f(t)$, $t \geq 0$ be a function such that $0 \leq m \leq f(t) \leq M < +\infty$. Consider $(i(t), s(t))$ that is solution of the following system of differential equations, with $\beta > 0$ and $i(0), s(0) \geq 0$,*

$$\begin{array}{rcl} \frac{d}{dt}i(t) & = & \beta i(t)s(t) \\ \frac{d}{dt}s(t) & = & -\beta s(t)i(t) - f(t)s(t). \end{array} \tag{37}$$

Then, we have the following estimates:

1) *$i(t) \leq i^+(t)$ and $s(t) \geq s^+(t)$, for all $t \geq 0$, where $(i^+(t), s^+(t))$ is solution of (37) with $f(t)$ replaced by m, whenever $i(0) \leq i^+(0) < +\infty$ and $s(0) \geq s^+(0) \geq 0$.*
2) *$i(t) \geq i^-(t)$ and $s(t) \leq s^-(t)$, for all $t \geq 0$, where $(i^-(t), s^-(t))$ is solution of (37) with $f(t)$ replaced by M, whenever $i(0) \geq i^-(0) \geq 0$ and $s(0) \leq s^-(0) < +\infty$.*

Proof. From (37),

$$\frac{d}{dt}(i(t) + s(t)) = -f(t)s(t).$$

Integrating, we have

$$i(t) + s(t) = i(0) + s(0) - \int_0^t f(u)s(u)du.$$

Plug-in $s(t)$ from the last identity into the first differential equation in (37) to obtain

$$\frac{d}{dt}i(t) = \beta i(t)\left(i(0) + s(0) - \frac{1}{\beta}\int_0^t f(u)s(u)du - i(t)\right). \tag{38}$$

Noting that $d\log i(t) = \beta s(t)$ and $d\log i(t) = di(t)/i(t)$ transform the differential equation in the last display into

$$\frac{d}{dt}\log i(t) = \beta\left(i(0) + s(0) - \int_0^t f(u)d\log i(u) - i(t)\right).$$

Now, $d\log i(t) \geq 0$, so that

$$\begin{array}{rcl} \frac{d}{dt}\log i(t) & \leq & \beta\Big(i(0) + s(0) - m\int_0^t d\log i(u) - i(t)\Big) \\ & = & \beta\Big(i(0) + s(0) - m\log\frac{i(t)}{i(0)} - i(t)\Big). \end{array}$$

The last inequality permits us to use standard comparison argument to conclude that $i^+(t)$, defined as the solution of the differential equation specified by taking equality in the last relation, is a super-solution for $i(t)$, which shows the first assertion of the result. In view of the second differential equation in (37), $i(t) \leq i^+(t)$, $t \geq 0$, implies $(d/dt)s(t) \geq -s(t)i^+(t) - f(t)s(t)$, so that by a similar comparison argument, $s^+(t)$ is sub-solution for $s(t)$. This proves item 1 of the lemma. The item 2 follows similarly, which completes the proof of the lemma.

We now proceed with the proof of the proposition. The proof is by induction showing that under condition C, the bounds established in Lemma 1 propagate over time.

Fix an interval $[0, t]$. By hypothesis, there exists a finite partition $0 \leq t_1 \leq \cdots \leq t_{n-1} \leq t_n = t$, so that condition C holds. Base step: under C, there exists m_0 such that $f_1(u) \geq m^0 \geq f_2(u)$, for all $u \in [0, t_1]$, hence by Lemma 1, $i_1(u) \geq$

$i_2(u)$ and $s_1(u) \le s_2(u)$, for all $u \in [0, t_1]$. Inductive step: assume $i_1(u) \ge i_2(u)$ and $s_1(u) \le s_2(u)$, for all $[0, t_m]$, for some $m < n$. Then, from Lemma 1, $i_1(u) \ge i_2(u)$ and $s_1(u) \le s_2(u)$, for all $u \in [t_m, t_{m+1}]$. The proof follows by induction.

D. Proof of Theorem 2

We consider the patching system (7–8). By definition of $w(t)$, it holds $0 \le w(t) \le 1$, all $t \ge 0$. By the hypothesis of the theorem, $1 \le a(t)$, for all $t \ge T$, so that by definition of $w(t)$, we have $w(t) = 1$, for all $t \ge T$. It follows that $w(t) \ge 1_{t \ge T}$, $t \ge 0$.

It is readily checked that hypotheses of Proposition 2 hold, so that by setting $i_2(t) \equiv i(t)$ and $f_1(t) \equiv 1_{t \ge T}$ and $f_2(t) \equiv a(t)$, we have $i(t) \le i_1(t)$, under $i_1(0) = i(0)$. In view of the representation (38) of an equivalent system, we can directly write

$$\frac{d}{dt} \log i_1(t) = \begin{cases} \beta(i(0) + s(0) - i_1(t)), \ 0 \le t < T \\ \beta \left(i_1(T) + s_1(T) - \frac{\mu}{\beta} \log \frac{i_1(t)}{i_1(T)} \right. \\ \left. - i_1(t) \right), \ t \ge T. \end{cases}$$

The solution on $[0, T]$ can be obtained in closed form as the differential equation on $[0, T]$ is standard logistic. Plugging this solution for $i_1(T)$ and noting that indeed $i_1(T) + s_1(T) = i(0) + s(0)$, yields, for any $t \ge T$,

$$i_1(t) + \frac{\mu}{\beta} \log \left(\frac{i_1(t)}{i(0)} \frac{i(0) + s(0) e^{-\beta(i(0)+s(0))T}}{i(0) + s(0)} \right) \le i(0) + s(0).$$

The left-hand side is increasing with $i_1(t)$, so that replacing $i_1(t)$ with $i(t) \le i_1(t)$, we have, for $t \ge T$,

$$i(t) + \frac{\mu}{\beta} \log \left(\frac{i(t)}{i(0)} \frac{i(0) + s(0) e^{-\beta(i(0)+s(0))T}}{i(0) + s(0)} \right) \le i(0) + s(0).$$

The proof follows.

E. Proof of Theorem 3

From Theorem 1, and $d \log i(t) = \beta s(t)$, we have

$$\mu \int_0^{+\infty} w(t) s(t) dt = i(0) + s(0) - i(+\infty).$$

This can be rewritten as

$$\frac{\mu}{\beta} \log \frac{i(+\infty)}{i(0)} - \mu \int_0^{+\infty} (1 - w(t)) s(t) dt = i(0) + s(0) - i(+\infty). \tag{39}$$

From (10), for any fixed $t \ge 0$,

$$\mu(1 - w(t)) \sim \mu(1 - w(0)) \frac{1}{1 - a(0) + a(0) e^t}, \text{ as } \mu \to 0.$$

Furthermore, for any fixed $t \ge 0$, $s(t) \sim s(0)$, as $\mu, \beta \to 0$. In view of (39), the result follows by integrating

$$\int_0^{+\infty} \frac{dt}{1 - a(0) + a(0) e^t} = \frac{1}{1 - a(0)} \log \left(\frac{1}{a(0)} \right).$$

F. Patching and Filtering (25)–(26), the Many-Subnets Limit

Whenever a superhost j is alerted, i.e. $V_j(t) = 1$, a worm scan originating from or destined to the subnet of superhost j fails with probability 1. In other words, a worm scan at time t from subnet k to subnet m succeeds only if both $V_k(t) = 0$ and $V_m(t) = 0$. As earlier, each host under an alerted superhost installs a patch with rate μ.

The numbers of susceptible and infected hosts are specified by a Markov process with the following transition rates:

$$\begin{array}{ll} I_j \to I_j + 1: & \frac{\eta}{\Omega} \left(I_j + \sum_{i \ne j} I_i (1 - V_i)(1 - V_j) \right) \\ S_j \to S_j - 1: & \frac{\beta}{\Omega} \left(I_j + \sum_{i \ne j} I_i (1 - V_i)(1 - V_j) \right) - \mu V_j S_j, \\ V_j \to V_j + 1: & \frac{1}{J} \left(\sum_{k=1}^{J} V_k \right) (1 - V_j). \end{array}$$

Assume now that the number of vulnerable hosts in any subnet is bounded, i.e. there exists a fixed $M > 0$, so that for the number of vulnerable hosts N_j, holds $N_j \le M$, for any subnet j. We define $K_{i,j}$ as the number of dormant subnets with i infected hosts and j susceptible hosts. The process $\mathbf{K} := (K_{i,j})_{i,j}$ is Markov with transition rates:

$$\mathbf{K} \to \begin{cases} \mathbf{K} + e_{i,j} - e_{i-1,j+1} & : \frac{\eta}{\Omega} \left(\sum_{n,m} n K_{n,m} \right) (j+1) K_{i-1,j+1} \\ \mathbf{K} - e_{i,j} & : \frac{1}{J} \left(J - \sum_{n,m} K_{n,m} \right) K_{i,j} \end{cases}$$

where $e_{i,j}$ is a $M \times M$ matrix with the (i, j) element equal to 1 and other elements equal to 0. Note that $I_D = \sum_{i,j} i K_{i,j}$ and that the transition rates of I_D are given by:

$$I_D \quad \to \quad \begin{cases} I_D + 1 & \frac{\eta}{\Omega} \left(\sum_{i,j} i K_{i,j} \right) \sum_{i,j} j K_{i,j} \\ I_D - n & \frac{1}{J} \left(J - \sum_{i,j} K_{i,j} \right) \sum_j K_{n,j} \end{cases}$$

Assume now that $\beta = \eta J / \Omega$ is fixed as J tends to infinity, and note that $\mathbf{K}$ is a density-dependent Markov process. Similarly as in Section A, the Kurtz's convergence result yields that the scaled version of (I_D, S_D) converges to (25)–(26) as J goes to infinity.

G. Proof of Theorem 4

Observe from (25) and (26) that

$$\frac{d}{dt}(i_D(t) + s_D(t)) = -(i_D(t) + s_D(t)) a(t),$$

and so,

$$i_D(t) + s_D(t) = (i_D(0) + s_D(0)) e^{-\int_0^t a(u) du}. \tag{40}$$

Plugging $s_D(t)$ obtained from above into (25), we obtain the generalized logistic equation

$$\frac{d}{dt} i_D(t) = \beta i_D(t) (f(t) - i_D(t)) \tag{41}$$

with

$$f(t) = (i_D(0) + s_D(0)) e^{-\int_0^t a(u) du} - \frac{1}{\beta} a(t). \tag{42}$$

We shall use the following general result for a generalized logistic equation.

Proposition 4. *Consider the generalized logistic equation,*

$$\frac{d}{dt} y(t) = \beta y(t) (f(t) - y(t)), \quad \beta > 0. \tag{43}$$

Define $F(t) = \int_0^t f(u) du$, and suppose that $f(t)$ is a real-valued function for which $F(t) < \infty$ for all $t \ge 0$. Then, the

solution of (43) is given by

$$y(t) = y(0)\frac{e^{\beta F(t)}}{1+\beta y(0)\int_0^t e^{\beta F(u)}du},\ t \geq 0. \qquad (44)$$

Proof: Let $Y(t) = \int_0^t y(u)du$. Equation (43) can be rewritten as

$$\frac{d}{dt}\log y(t) = \beta f(t) - \beta y(t).$$

It follows that

$$\log\frac{y(t)}{y(0)} = \beta F(t) - \beta\int_0^t y(u)du.$$

Hence,

$$Y'(t) = y(t) = y(0)e^{\beta F(t)}e^{-\beta Y(t)}. \qquad (45)$$

This differential equation can be solved explicitly by the classical method of separation of variables. Indeed,

$$e^{\beta Y(t)}dY(t) = y(0)e^{\beta F(t)}dt.$$

Integrating, we have

$$e^{\beta Y(t)} = 1 + \beta y(0)\int_0^t e^{\beta F(u)}du,$$

since $e^{\beta Y(0)} = 1$. In view of the last identity and (45), the claim of the proposition follows.

We shall now use this result to solve the particular generalized logistic equation (41). Assume without loss of generality that $i_D(0) + s_D(0) = 1$. Using (6) and (36), we can rewrite (42) as

$$f(u) = \frac{i_D(0)+s_D(0)}{1-a(0)+a(0)e^u} - \frac{a(0)e^u}{\beta[1-a(0)+a(0)e^u]}. \qquad (46)$$

Next, making the change of variables $x = e^u$, we get

$$\begin{aligned}\beta F(t) &= \int_1^{e^t}\frac{\beta(i_D(0)+s_D(0))dx}{x[1-a(0)+a(0)x]} - \frac{a(0)dx}{1-a(0)+a(0)x}\\ &= \beta' t - \log\left(1-a(0)+a(0)e^t\right)^{1+\beta'}\end{aligned}$$

where $\beta' := \beta(i_D(0)+s_D(0))/(1-a(0))$. Hence, in view of (6), we have

$$e^{\beta F(t)} = e^{-t}\Big(\frac{a(t)}{a(0)}\Big)^{1+\beta'}. \qquad (47)$$

We now proceed with computing $\int_0^t e^{\beta A(u)}du$. Making the change of variables $x = a(0) + (1-a(0))e^{-t}$, we obtain

$$\begin{aligned}\int_0^t e^{\beta F(u)}du &= \frac{1}{1-a(0)}\int_{a(0)+(1-a(0))e^{-t}}^1 \frac{dx}{x^{1+\beta'}}\\ &= \frac{1-a(0)}{\beta'}\Big(\Big(\frac{a(t)}{a(0)}\Big)^{-\beta'} - 1\Big). \qquad (48)\end{aligned}$$

Now, substituting (47) and (48) in (44), we get

$$i_D(t) = i_D(0)\frac{\left(\frac{a(t)}{a(0)}\right)^{1+\beta'}}{\frac{s_D(0)}{i_D(0)+s_D(0)} + \frac{i_D(0)}{i_D(0)+s_D(0)}\left(\frac{a(t)}{a(0)}\right)^{\beta'}}e^{-t}.$$

After appropriate substitutions, this is the same as (28). The asserted identity for $s_D(t)$ follows in the view of (40), which completes the proof.

PLACE PHOTO HERE

Milan Vojnović is a researcher with systems and networking group at Microsoft Research Cambridge, United Kingdom. His research interests are in architecture and performance of networks and computer systems; in particular, wireless & mobile systems, congestion control, and information dissemination systems. He received his PhD in Communication Systems from EPFL, Switzerland, in 2003, and both MSc and BSc in Electrical Engineering from the University of Split, Croatia, in 1998 and 1995, respectively. He received ACM SIGMETRICS 2005 Best Paper Award (with Laurent Massoulié) for a work on performance of file swarming systems, IEEE INFOCOM 2005 Best Paper Award (with Jean-Yves Le Boudec) for a work on stationarity and perfect simulation of random mobility models, and ITC-17 2001 Best Student Paper Award (with Jean-Yves Le Boudec) for a work on TCP-friendliness of equation-based congestion control. In 2005, he has been awarded ERCIM Cor Baayen Award.

PLACE PHOTO HERE

Ayalvadi J. Ganesh graduated from the Indian Institute of Technology, Madras, in 1988, and received his MS and PhD in Electrical Engineering from Cornell University in 1991 and 1995. He has been with Microsoft Research, Cambridge, since 1999. His research interests include large deviations, random graphs and queueing theory, and applications to the Internet and P2P and wireless networks.