# Information Survival Threshold in Sensor and P2P Networks

Deepayan Chakrabarti[*], Jure Leskovec[†], Christos Faloutsos[†], Samuel Madden[‡],
Carlos Guestrin[†] and Michalis Faloutsos[§]
[*]Yahoo Research, Email: deepay@yahoo-inc.com
[†]Carnegie Mellon University, Email: {jure, christos, guestrin}@cs.cmu.edu
[‡]MIT, Email: madden@csail.mit.edu
[‡]University of California, Riverside, Email: michalis@cs.ucr.edu

***Abstract*— Consider a network of, say, sensors, or P2P nodes, or bluetooth-enabled cell-phones, where nodes transmit information to each other and where links and nodes can go up or down. Consider also a 'datum', that is, a piece of information, like a report of an emergency condition in a sensor network, a national traditional song, or a mobile phone virus. How often should nodes transmit the datum to each other, so that the datum can survive (or, in the virus case, under what conditions will the virus die out)? Clearly, the link and node fault probabilities are important — what else is needed to ascertain the survivability of the datum?**

**We propose and solve the problem using non-linear dynamical systems and fixed point stability theorems. We provide a closed-form formula that, surprisingly, depends on only one additional parameter, the largest eigenvalue of the connectivity matrix. We illustrate the accuracy of our analysis on realistic and real settings, like mote sensor networks from Intel and MIT, as well as Gnutella and P2P networks.**

## I. Introduction

In this work, we focus on the conditions under which a self-replicating object can survive in an unreliable network. We assume a network (e.g. a sensor network) where initially some nodes have an object (e.g. a query, or some other datum). Nodes and edges are unreliable: edges may be up or down, and with some probability, nodes may die (e.g., run out of batteries); we also assume that they then resurrect with some resurrection rate (e.g., someone installs fresh batteries). We assume that a node loses the object in case of death and subsequent resurrection. (Our upcoming analysis could be easily modified to handle the converse assumption, but this is outside the scope of this work). With some transmission probability, the object may be transmitted from a node that has it, to its neighbors; if the link and the neighbor are "up" at the time, the copy is successful.

For example, consider a cellphone network where the communication between nodes is subject to loss (link failures), and nodes may go down (battery failure, shut down by user or moved out of range). Consider some static piece of information, or "datum", such as a mobile phone virus. As cellphones go down, the virus dies; however, when the cellphone is up and infected, the virus infects other phones. We seek the conditions under which the virus would die out and not become an epidemic. Conversely, we could have some information (say, an emergency alert, or an important "reading") in a sensor network, and we want this information to survive. In a high failure-rate environment (e.g., fire or evacuation system), there might be very little time between detection of the event and the destruction of the node. We want to get the data off the node as quickly as possible and spread it through the network so that the information will survive.

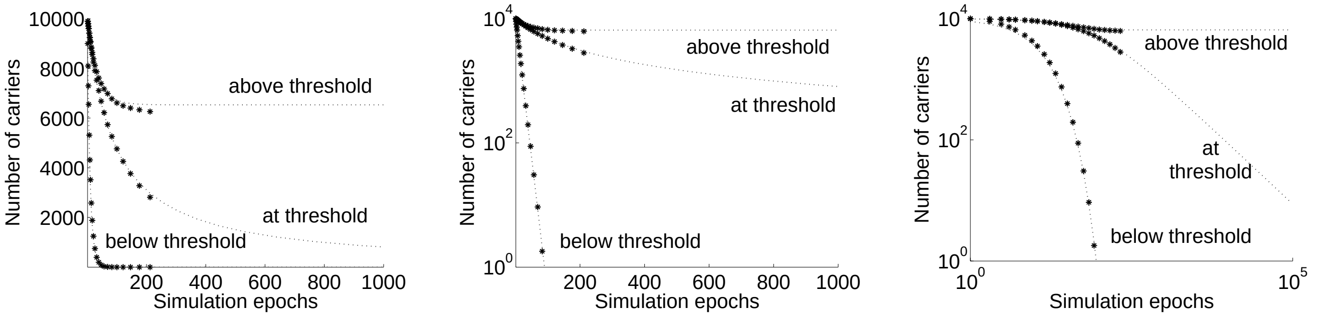
Thus, informally, the problem can be stated as follows:

> *Under what conditions can we expect the object or datum (e.g. the virus, or the piece of information) to survive or die out in a dynamic network?*

We can identify two major cases. In the first, if the transmission rate is not fast enough, the object will eventually disappear from the network. In the second case, if the transmission rate is fast enough, then the datum will take over a significant part of the network[1] and it will linger practically for ever. Interestingly, there is a fascinating, and sharp, *phase transition* between the two regimes. The next example illustrates the above concepts.

Example: Figure 1 shows an example of information survival on a 2D-grid graph with $N = 10,000$ nodes (Section IV has more details; the qualitative behavior of real graphs is similar). For each time instant $t$, we plot the number of "carriers", that is the number of 'up' nodes carrying the datum. We plot in linear-linear, log-linear, and log-log scales (plots (a-c), respectively), to illustrate the qualitative difference in each regime. Using our upcoming analysis, we chose three sets of parameters settings, so that we are below-, at-, and above- the threshold. As we can see, there is a significant, qualitative difference in behavior under these three settings. Below the threshold, the information dies out *exponentially quickly*. This is shown on figure 1(b), where the exponential function becomes a straight line when plotted on log-linear scales. Exactly at the threshold (and by "exactly" we mean several significant digits!), the number of carriers decays *polynomially*, following a power law $t^a$ and becoming a straight line on log-log scales (see Figure 1(c)). Above the threshold, the expected number of carriers stabilizes at a non-zero value, and the information lasts practically for ever, although the number of carriers may

[1] The datum will not necessarily take over the network completely, since nodes are continuously dying and waking up, so there are always some nodes that are alive but without the information.



(a) carriers vs. time (lin-lin)

(b) same, in log-lin

(c) same, in log-log

Fig. 1. *Survival of information on a grid network.* Number of carriers, over time, for a 2-D grid network with $N$=10,000 nodes, and for below, at and above (our) threshold. Notice very different qualitative behaviors: below threshold, the information dies out *exponentially* quickly (line, in log-lin scales, of Figure (b)). Exactly at threshold, the information is dying out as a power law (line, in log-log scales, of Figure (c)). Above threshold, the information survives, practically for ever (all figures).

be less than $N$, because of down-nodes, and not-yet-infected nodes. We give a precise definition of the term "practically for ever" in Section III.

Our work provides an analytical model and identifies the fundamental conditions for which a datum will survive or become extinct in a dynamic network. Although the problem definition is deceptively simple, there is no known practical, exact solution. The obvious one, discussed later in Section III, requires Markov Chains, and is prohibitively expensive: its computational cost is proportional to $3^N$, where $N$ is the number of nodes. For reference, $3^{200}$ is comparable to the number of electrons in the universe. Note that our model is very general since it considers: (a) a dynamic network (with failing links and nodes), (b) arbitrary topology, and (c) does not make any assumption about the initial conditions.

*Our Contributions:* They are the following.

1) *Closed form formula:* Based on our novel dynamical system model, we derive a survivability condition that is *extremely* simple, general and accurate. In fact, our formula includes as a special case the SIS model of viral propagation (described later).
2) *Experimental validation of our model:* Extensive simulations on several realistic topologies (sensor from Intel, MIT, and P2P networks like Gnutella) show that our model is highly accurate in determining the behavior of the system and identifying the threshold for the phase transition.

*Our work in perspective.* In addition to its theoretical merit, our survivability condition provides a starting point for: (a) network design, and (b) interpretation of empirical or simulated results. First, network designers would use our results to choose the node density, the network size, or the datum retransmission frequency in order to save or drive to extinction the datum, when these parameters are known or under the control of the designer. Second, even when we do not know or cannot control these properties, our fundamental relationships will help researchers interpret the observed or simulated results.

Our work was initially motivated by sensor and P2P network design, but it is also applicable in several other settings:

- Virus containment and anti-virus protection, where the "datum" is a virus: Here, we *do* want to drive the datum to extinction, and our approach allows us to decide how often to *quarantine* each node and for how long.
- Social networks: News, rumors, and web-log ("blog") dissemination, marketing and fad propagation, and many more applications that seek to propagate and maintain information can be handled under this framework.

The remainder of the paper is organized as follows: Section II surveys the related work. Section III describes our non-linear dynamical system approach, and gives the major theorems and proofs. Section IV gives experimental results. We conclude in Section V.

## II. Related Work

Graphs and sensor networks have attracted a lot of interest lately, for quick and efficient aggregation of information [20], [11], for understanding "trust" and "distrust" in online social networks [21], and in several other areas. With respect to our problem, the closest related work has been in the areas of gossip-based protocols, epidemiology, and computer security

*a) Gossip-based protocols:* Gossip-based protocols have been studied in both peer-to-peer as well as sensor and other ad-hoc networks, where nodes may be up or down. The rate of node turnover in such networks is referred to as *churn* [40]; in high-churn settings, gossiping is often useful as a mechanism for ensuring eventual-consistency of state in distributed networks despite unpredictable node populations and connectivity.

Thus, gossip-based protocols have been studied in high-churn cases, for reliable multicast and broadcast protocols [33], [23], [36], [4], resource location [30], [38], failure detection [46], [39], [42], [7], [50], database aggregation [28], database and peer-to-peer replication [12], [10], and ensuring the stability of dynamic hash table-based peer-to-peer systems [22], [40]. Information dissemination under memory constraints have also been studied [35]. Empirical and theoretical studies of gossip protocols include [5], [16], [33], [30], [29], [49].

However, they all assume that the initial infection or rebroadcast rate is high enough that dying out is not a concern. In this work we exactly quantify the conditions for survivability.

*b) Epidemiology:* The epidemiology community has developed the so-called *SIR* and *SIS* models [2] of infection. The *SIS* model (*Susceptible – Infective – Susceptible*) is suitable for, e.g., the common flu, where nodes may be infected, healed (and susceptible), and infected again. The *SIR* model (*Susceptible – Infective – Removed*) is suitable for, say, mumps, where a node, after being infected, becomes removed (with life-time immunity).

The area of "interacting particle systems" is also remotely related: "particles" propagate over a simple network according to different processes; the one closest to our work is the "contact process" [24], [34], [13]. However, most previous work in this area assumes networks with (a) infinite size, and (b) regular topologies such as line graphs and grids.

The approach we present here is based on the SIS model – a node is "susceptible" to a data item when it is online and functioning normally; as nodes crash, they become "immune" for the duration of their failure, and later become "susceptible" again when they are back online. Intuitively, the model we focus on resembles an SIS model with random "quarantine". Our novelty is that we study *arbitrary* graph topologies and we are the first to derive the survivability condition for such cases.

*c) Computer security:* There are numerous studies of worm and virus propagation on the Internet [48], [32], [37], [43], [45], based on the *SIS*, *SIR* and *influence* models of infection [31], [8], [1], [19]. Others have done detailed forensic studies of the spread of worms [37], [43], [44] illustrating the exponential spreading predicted by SIR and SIS models, with the entire susceptible population quickly become infected and then slowly being "removed" as patches are applied. Epidemiological models have again been used in developing good quarantining strategies for scanning worms [17]. Worm propagation has been studied under special cases, such as in email networks [51] and on the IPv6 Internet [3]. Mathematical modeling of propagation behavior [47], [18] has provided some answers on "epidemic thresholds"; we show that our current work includes these results as a special case.

## III. Proposed Method

We are given a network of $N$ nodes (sensors, computers or people) and $E$ directed links between them. For ease of exposition, we assume discrete time-steps of size $\Delta t$, where $\Delta t$ is vanishing ($\Delta t \rightarrow 0$). The continuous-time version is omitted for space, because it gives identical survivability results.

Within a $\Delta t$ time interval, each node $i$ has a probability $r_i$ of trying to broadcast its information, and each link $i \rightarrow j$ has probability $\beta_{ij}$ of being "up", and thus correctly propagating the information to node $j$. Each node $i$ also has a node failure probability $\delta_i > 0$ (e.g., due to battery failure of the sensor). Every dead node $j$ has probability $\gamma_j$ of resurrecting to the "up" state, but without any information in its memory (e.g., due to the periodic replacement of dead batteries). The symbols we use are listed in Table I.

TABLE I

Table of symbols

| Symbol | Description |
|---|---|
| $N$ | Number of nodes in the network |
| $\beta_{ij}$ | Probability that link $i \rightarrow j$ is up |
| $\delta_i$ | Death rate: Probability that node $i$ dies |
| $\gamma_i$ | Resurrection rate:<br>Probability that node $i$ comes back up |
| $r_i$ | Retransmission rate:<br>Probability that node $i$ broadcasts |
| $p_i(t)$ | Probability that node $i$ is alive at time $t$ and has info |
| $q_i(t)$ | Probability that node $i$ is alive at time $t$ but without info |
| $1 - p_i(t) - q_i(t)$ | Probability that node $i$ is dead |
| $\nu_i(t)$ | Probability that node $i$ does *not* receive info from *any* of its neighbors at time $t$ |
| $\vec{\mathbf{p}}(\mathbf{t})$, $\vec{\mathbf{q}}(\mathbf{t})$ | Probability column vectors |
| $\bar{C}(t)$ | True number of carriers at time $t$ |
| $\hat{C}(t)$ | Estimated number of carriers at time $t$ |
| $C_{\text{“}\infty\text{”}}$ | Number of carriers at quasi-steady-state |
| $S$ | The $N \times N$ *system matrix* |
| $\lambda_{1,S}$ | The largest eigenvalue of $S$ |
| $s = \lvert\lambda_{1,S}\rvert$ | "Survivability score" |

This system can be modeled as a Markov chain, where each node can be in one of three states: "Has Info", "No Info" or "Dead", with transitions between them as shown in Figure 2. The full state of the system at any instant consists of $N$ such states, one for each node. Thus, there are $3^N$ system states. Transitions out of the current system state depend *only* on the current state and not on any previous states; thus it is a Markov chain.

There is an extremely subtle point here: observe that there is an absorbing set of states (where no node is in "Has Info") and that this set can be reached from any starting state. Thus, the information will die out with probability 1 as time tends to infinity (see [6]). However, from a practical point of view, when the parameter values are within a particular region of the parameter space, this extinction happens quickly. Defining this region in the parameter space is exactly the goal of our work. Outside this region, the time to extinction can be very long. For example, consider the case of the SIS model, which is a special case of our problem as we show in Corollary 2 in Section III-D: even for a simple line graph, under the SIS model and with above-threshold condition, the expected time to extinction $\tau$ grows *exponentially* with the size of the graph $N$. Specifically, $\tau \rightarrow c \cdot e^N$ as $N \rightarrow \infty$ [14]. As an arithmetic example, suppose that we are above threshold, on a network with $N$=1000 nodes, and that the time-tick is $\Delta t = 10^{-9}$ (the cycle time of a 1GHz processor). Then the expected time to extinction is $O(e^{1000} \times 10^{-9}) \approx 10^{417}$ years, while the age of the universe is of the order of billion ($10^9$) years. In such cases, the datum practically survives for "ever".

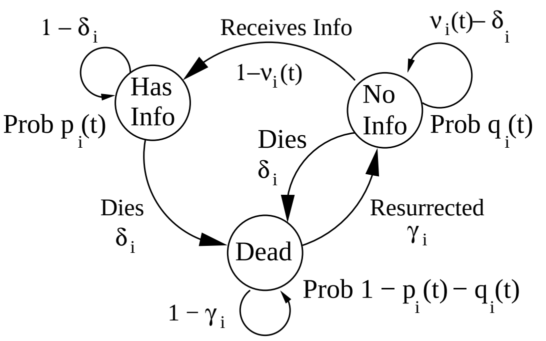


Fig. 2. *Transitions for each node:* This shows the three states for each node, and the probabilities of transitions between states.

The question is: under what conditions does the information survive for a long time, and when will the information die out quickly? Let $\bar{C}(t)$ denote the expected number of carriers (nodes in "Has Info" state) at time $t$. In general, $\bar{C}(t)$ decays exponentially, polynomially or logarithmically (with expected time to extinction comparable to or larger than the age of the universe for large graphs), depending whether the system is below, at or above the threshold [13], [14], [15]. Figures 1(a)-(c) illustrate these three cases. We focus on the fast extinction case, since many other works have looked at the rapid spread case (e.g., [37], [43]).

*Definition 1 (Fast Extinction):* "Fast extinction" (henceforth "extinction", for brevity) is the setting where the number of carriers $\bar{C}(t)$ decays exponentially with time ($\bar{C}(t) \propto c^{-t}, c > 1$).

We shall use the term "survival" for the converse case, where the time to extinction is astronomically high. We shall use the term "at the threshold" for the extremely improbable case when the number of carriers $\bar{C}(t)$ decreases as a power law with time. Finally, we shall use the term quasi-steady-state for the situation when we are above threshold and number of carriers seems stable, like the 'above threshold' case of Figure 1 for time-tick 200 and above. The number of carriers $C_{``\infty"}$ at the quasi-steady-state will be referred to as "*residual carriers*".

We can now formally state our problem:

*Problem 1: Given:* the network topology (link "up" probabilities) $\beta_{ij}$, the retransmission rates $r_i$, the resurrection rates $\gamma_i$ and the death rates $\delta_i$ $(i = 1 \ldots N, j = 1 \ldots N)$

*Find:* the condition under which a datum will suffer "fast extinction".

### A. Main Idea

Solving this problem for the full Markov chain requires $3^N$ variables and is thus intractable, even for moderate-sized networks. Exact values for the "fast extinction" threshold are unavailable even for simpler versions of this problem [18].

Our main contribution is an accurate approximation, using a non-linear dynamical system of only $N$ variables. Let $p_i(t)$ and $q_i(t)$ be the probabilities of node $i$ being in the "Has Info" and "No Info" states at time $t$, respectively. Thus, the probability of the node being dead is $(1 - p_i(t) - q_i(t))$. Then, we can approximate our setting with the following dynamical system:

*Lemma 1 (Dynamical system):* The probabilities $p_i(t)$ and $q_i(t)$ for node $i$ ($i = 1, \ldots, N$) to be in state "Has Info" and "No Info", respectively, at time $t$, are approximated by

$$\begin{aligned} p_i(t) &= p_i(t-1) * (1 - \delta_i) \\ &\quad + q_i(t-1) * (1 - \nu_i(t)) \;\; \forall i, \; \forall t \qquad (1) \\ q_i(t) &= q_i(t-1) * (\nu_i(t) - \delta_i) \\ &\quad + \gamma_i \left(1 - p_i(t-1) - q_i(t-1)\right) \;\; \forall i, \; \forall t \qquad (2) \end{aligned}$$

where $\nu_i(t)$ is the probability that node $i$ does *not* receive the information from any of its neighbors at time $t$, and it is given by

$$\nu_i(t) = \Pi_{j=1}^{N} \left(1 - r_j \beta_{ji} p_j(t-1)\right) \qquad (3)$$

*Proof:* Starting from state "No Info" at time $t-1$, node $i$ can acquire this information (and move to state "Has Info") if it receives a communication from some other node $j$. Let $\nu_i(t)$ be the probability that node $i$ does *not* receive the information from *any* of its neighbors. Then, assuming that the neighbors' states are *independent*, we use the transition matrix in Figure 2 and apply it for each node $i$, and write down the probabilities of being in each state at time $t$, *given* the probabilities at time $t-1$. Recall that we use time-steps of vanishing size $\Delta t$, exactly so that the probability of two events happening within the same time-tick is vanishingly small, and thus we can neglect second- and higher-order terms. Eq. 2 is derived though similar reasoning. ■

The reader may be skeptical about the impact of the independence assumption. However, as we show in Section IV, the assumption (Eqs. 1-3) leads to extremely accurate results for all the real and synthetic networks we tried. In fact, the dotted lines in Figure 1(a-c) all correspond to the estimations with the *Dynamical System* and the independence assumption, while the black circles correspond to averages, after we run simulations; notice how close the results are.

Next, we discuss the properties of this dynamical system, and specifically we study the condition for fast extinction on this system.

### B. Main Result

Our goal is to find the conditions under which we have "fast extinction". The high-level description of our approach is the following: (a) We start from the Dynamical System equations (Eq (1)-(3)), (b) we show that it has a fixed point (namely, when the datum/virus is extinct), and (c) we find the conditions under which this fixed point is "stable". Under exactly those conditions, the system will quickly return to the extinct state.

We present the details next. After appropriately manipulating the Dynamical System equations (described in the extended version [9]), we get the so-called *system matrix* $S$, which is pivotal for the rest of the analysis. This is an $N \times N$ square matrix, defined as follows:

*Definition 2 (System Matrix):*

$$S_{ij} = \begin{cases} 1 - \delta_i & \text{if } i = j \\ r_j \beta_{ji} \frac{\gamma_i}{\gamma_i + \delta_i} & \text{otherwise} \end{cases} \quad (4)$$

for $i = 1, \dots N$, and $j = 1, \dots N$.

Intuitively, the diagonal of the matrix has the terms $1 - \delta_i$, which give the probability that $i$-th node will remain alive. The off-diagonal elements $S_{ij}$ of the matrix contain the probability that node $i$ will be infected by node $j$: $\frac{\gamma_i}{\gamma_i + \delta_i}$ is the probability node $i$ is alive and without the information, $r_j$ is the probability that $j$ transmits information and $\beta_{ji}$ the probability that the transmission will succeed.

Let $|\lambda_{1,S}|$ be the magnitude of the largest eigenvalue (in magnitude).

*Definition 3 (Survivability score):* The largest eigenvalue $s = |\lambda_{1,S}|$ of the system matrix $S$ is defined as "survivability score" for the system.

Let $\hat{C}(t)$ to be the expected number of carriers at time $t$ according to this dynamical system; $\hat{C}(t) = \sum_{i=1}^{N} p_i(t)$.

*Theorem 1 (Condition for fast extinction):* If the survivability score $s = |\lambda_{1,S}|$ obeys

$$\boxed{s = |\lambda_{1,S}| < 1}$$

then we have fast extinction in the dynamical system, that is, the expected number of carriers $\hat{C}(t)$ decays exponentially over time.

*Proof:* The proof follows from Lemma 2 and Theorems 2 and 3. For the full details, see the extended version [9]. At the high level, the proof examines the stability of the fixed point of Eqs. 1,2. The fixed point is the case where no node carries the datum ($p_i(t) = 0 \ \forall i$). A dynamical system has a stable fixed point if the first eigenvalue of the Jacobian matrix at that point is smaller than 1. In our case, the first eigenvalue of the Jacobian matrix is exactly the same as that of the *System Matrix* of Eq. 4. ■

*Definition 4 (Threshold):* We will use the term "below threshold" when $s < 1$, "above threshold" when $s > 1$, and "at the threshold" for $s = 1$.

The results above are very general, and, as we show via simulations, very accurate as well. Next, we examine one common special cases, to demonstrate the intuitive behavior of the system.

*Corollary 1 (Homogeneous reliable-link case):* If all nodes exhibit similar behavior, $\delta_i = \delta, r_i = r, \gamma_i = \gamma$ for all $i$, and $B = [\beta_{ij}]$ is a symmetric binary matrix (links are undirected, and are always up or always down), then the condition for fast extinction is:

$$\frac{\gamma r}{\delta(\gamma + \delta)} \lambda_{1,B} < 1 \quad (5)$$

*Proof:* The system matrix $S$ can be written as $S = (I * (1 - \delta) + B * r \cdot \gamma/(\gamma + \delta))$ where $I$ is the $N \times N$ identity matrix. From the properties of eigenvalues, we have that $\lambda_{1,S} = (1 - \delta) + \lambda_{1,B} * r \cdot \gamma/(\gamma + \delta)$ and, combining with Theorem 1, we have the proof. ■

The above result agrees with intuition: The survivability of the datum increases with the connectivity $\lambda_{1,B}$, the retransmission rate $r$ and the resurrection rate $\gamma$; and decreases with the death rate $\delta$.

### C. Lemmas and other results

First, we show that the scenario with no information survival ($p_i(t) = 0$) forms a fixed point of the dynamical system. Then, we show that below the threshold condition of Theorem 1, this fixed point is *asymptotically stable* under small perturbations (this is how we derived the condition in Theorem 1). Finally, we show that our threshold is insensitive to the starting state: below the threshold, $p_i(t) \to 0$ and thus $\hat{C}(t) \to 0$ exponentially quickly. Detailed proofs are provided in extended version [9].

Before we give the formal version, we present the intuition. A dynamical system, like, e.g., a ball on a flat surface, has a fixed point if it is at equilibrium there. The fixed point is said to be stable (eg., a ball inside a spherical bowl, with non-zero friction, resting at its bottom), if the system returns to that point, despite a small perturbation. The ball inside a bowl is a dynamical system with a few variables (2 polar coordinates for the ball, and a few more variables for its velocity vector). In our case, we have 2*$N$ variables, the $p_i(t)$ and $q_i(t)$ for each node $i$. Our goal is to find the fixed point for our 2*$N$-dimensional vector, given the transitions equations (Eq. 1-3) and study the conditions under which this point will be stable.

*Definition 5 (Asymptotic Stability of a Fixed Point):* A fixed point $P_f$ is "asymptotically stable" if, on a slight perturbation from $P_f$, the system returns to $P_f$ (as opposed to moving away, or staying in the neighborhood of $P_f$ but not approaching it) [25].

Mathematically, this means that the Jacobian matrix of the system, computed at point $P_f$, has all eigenvalues smaller than 1 in magnitude [25].

*Lemma 2 (Fixed Point):* The values

$$\left( p_i(t) = 0, q_i(t) = \frac{\gamma_i}{\gamma_i + \delta_i} \right)$$

for all nodes $i$, are a fixed point of Eqs. 1-3.

*Proof:* By substitution into Equations 1-3. ■

*Theorem 2 (Stability of the fixed point):* The fixed point of Lemma 2 is *asymptotically stable* if the system is below the threshold, that is, $s = |\lambda_{1,S}| < 1$.

*Proof:* Omitted, for space (see [9]). The sketch of the proof is as follows: We compute the $2N \times 2N$ Jacobian of our dynamical system, and request that the largest eigenvalue magnitude $< 1$. It turns out that this is exactly the eigenvalue of the $N \times N$ *system matrix* that we defined earlier. ■

*Theorem 3 (Insensitivity to the starting state):* If we are below threshold ($s = |\lambda_{1,S}| < 1$), then we have fast extinction *regardless* of the starting state.

*Proof:* See extended version of the paper [9]. ■

### D. Corollaries and Special Cases

Here we present some special cases and corollaries, and show that the results agree with our intuition.

*Corollary 2:* We include the SIS model of viral infection as a special case.

*Proof:* The SIS model has only two states per node: "Has Infection" and "No Infection." In our model, if we increase the resurrection rate $\gamma$ so that a "dead" node comes back "up" very quickly, we can give the appearance of only *two* states: "Has Info" and "No Info", and thus mimic the SIS model. ■ In fact, if the ratio of resurrection-vs-death rate ($\gamma$ over $\delta_i$) increases to infinity, and all death rates are the same, the fast-extinction condition of Corollary 1 becomes $r/\delta \cdot \lambda_{1,B} < 1$ This is exactly the epidemic threshold condition for the SIS model [47], [18].

*Corollary 3 (P2P resilience):* Consider a *star* network (one hub and many satellite nodes) and a *ring* network (nodes in a circle) with the same number of nodes $N > 5$. They have similar number of edges ($N-1$ for *star*, $N$ for *ring*). However, the *star* network has higher higher survivability score, and the gap widens with the number of nodes $N$.

*Proof:* $|\lambda_{1,B_{star}}| = \sqrt{N-1} > 2 = |\lambda_{1,B_{ring}}|$. So, the *star* network has higher survivability score. ■

This agrees with intuition: in the *star* graph, the central node will have the datum/virus with very high probability, and it will keep transmitting it to the satellite nodes, infecting several of them, which will in turn infect it back later. In the *ring* network, every infected node has only two neighbors/chances to infect - if it fails, the system is one step closer to extinction. Again, we highlight the fact that the *star* network outperforms the *ring* on survivability, despite the fact that it is sparser by one edge.

## IV. Experiments

To verify our assumptions, we run a set of simulation experiments on several real and synthetic networks. We show that

1) Our Equations 1-3 accurately track the true dynamics of the system, and give excellent estimates $\hat{C}(t)$ for the number of carriers at time $t$;
2) The threshold condition derived in Theorem 1 is accurate and sharp; and
3) The final behavior of the system is insensitive to the starting conditions.

Next, we describe our datasets and simulation parameters, and then present the experimental results.

### A. Datasets

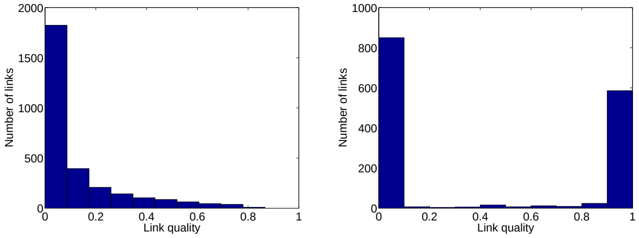
Four different datasets were used: These include one synthetic, one Peer-to-Peer and two sensor network deployment datasets. The datasets vary in both size as well as topology.

- *GRID*: This is a large synthetic 2D grid with $N$ = 10,000 nodes and $E$ = 39,600 edges. The link "up" probabilities ($\beta_{ij}$) are set to $0.1$ between all neighbors on the grid.

- *GNUTELLA*: This is a snapshot of the Gnutella peer-to-peer file sharing network, collected in March $2001$ [41], with $N$= 62,586 nodes and $E$ = 295,784 edges. The link "up" probabilities $\beta_{ij}$ are set to $0.1$ for the existing edges.

TABLE II
Parameter settings for the datasets.

| Dataset | threshold | $\delta$ | $\gamma$ | $r$ | $s$ |
|---|---|---|---|---|---|
| *GRID* | below | 0.1 | 0.01 | 0.1 | 0.90 |
| | at | 0.01 | 0.004 | 0.1 | 1.001 |
| | above | 0.01 | 0.1 | 0.1 | 1.02 |
| *GNUTELLA* | below | 0.1 | 0.01 | 0.1 | 0.91 |
| | at | 0.07 | 0.004 | 0.1 | 1.003 |
| | above | 0.01 | 0.01 | 0.1 | 1.05 |
| *INTEL* | below | 0.1 | 0.01 | 0.1 | 0.96 |
| | at | 0.02 | 0.0006 | 0.1 | 1.0003 |
| | above | 0.01 | 0.01 | 0.1 | 1.33 |
| *MIT* | below | 0.15 | 0.01 | 0.1 | 0.96 |
| | at | 0.05 | 0.0006 | 0.1 | 1.01 |
| | above | 0.01 | 0.01 | 0.1 | 1.88 |



(a) *INTEL* link qualities (b) *MIT* link qualities

Fig. 3. *Link quality distributions:* Plots (a) and (b) plots the number of links versus link quality. Pairs of sensors which cannot communicate with each other have a link quality of 0. While the *INTEL* distribution shows a broad range of link qualities, the *MIT* distribution is very highly peaked.

- *INTEL*: A $54$-node sensor network observed over a period of $33$ days [27]. The nodes are Mica2Dot sensors collecting time-stamped topology information once every 31 seconds. The data was collected using the TinyDB in-network query processing system, built on the TinyOS platform. The link-up probabilities $\beta_{ij}$ were estimated from the collected data. The nodes were deployed in a lab with a rectangular shape and "soft" walls which can be penetrated by radio signals, leading to a high average node degree ($\approx 46$) in the network. Figure 3(a) shows the distribution of link qualities $\beta_{ij}$, which are smeared-out over the entire range. The average link quality (considering only the links with non-zero link quality) is very low ($0.14$).

- *MIT*: This is a $40$-node sensor network at MIT (see [26] for an earlier version of the network). Each node is a Crossbow Mica2. Each node was attached to a Crossbow MIB600 interface board that provides both power and an Ethernet backchannel for programming and data collection. Sensors are placed in a elongated "corridor". This implies a lower average node degree ($\approx 18$); however, the link quality distribution is very peaked, figure 3(b). This leads to a high value of $0.92$ for the average link quality (again only considering the non-zero quality links). Note, these conditions are the exact opposite of what we see in the *INTEL* dataset.

### B. Accuracy of the dynamical system

For each network we set the parameters so that the system was below-, above- and at- threshold according to Theorem 1, as shown in Table II. Given the network and the estimated link qualities, we chose parameter values so that they are relatively close to the threshold.

We initialize all nodes in the "Has Info" state ($\bar{C}(t = 0) = N$), since the final state is insensitive to the initial conditions (see Theorem 3 and the experiments in Section IV-D). We then run the simulation for $T = 10,000$ steps, according to the state diagram of Figure 2, and we record the number of carriers $C(t)$ (nodes with information) for each of the $T$ epochs. Then, we repeated each simulation $100$ times and we record the average and standard deviation of the number of carriers at each epoch.

Figure 4 shows the number of carriers over time (only $200$ simulation epochs are shown for visual clarity; the results are similar over $T = 10,000$ timesteps). Simulation results are shown in solid lines, along with confidence intervals (+/- one standard deviation). We also ran our dynamical system (Equations 1-3) with exactly the same parameters and plot our estimated number of carriers $\hat{C}(t)$ in dotted lines. We observe the following:

• *The dynamical system is very accurate:* The dotted lines of our dynamical system are visually indistinguishable from the solid lines of the simulation (relative error is just around $1\%$). Thus, Equations 1-3 and their independence assumption are highly accurate for a wide variety of real-world settings.

• *The information dies out below our threshold:* For all the datasets, the number of carriers goes to zero very quickly below the threshold.

• *Above our threshold, the number of carriers remains practically constant:* For all the datasets, the information survives for a "long" time.

• *Variance decreases with network size:* Large networks, like *GRID* and *GNUTELLA*, had small variance, which makes the error bars invisible in Figures 4(a-b). Smaller networks, like the *INTEL* and *MIT* datasets show wider confidence intervals. In retrospect, this makes sense, probably being related to the law of large numbers.

### C. Accuracy of the threshold condition

In this set of experiments, we vary one parameter while keeping all the others fixed. The link qualities $\beta_{ij}$ depend on the environment, while the death rate $\delta$ is intrinsic to the sensor and its battery; thus, we only perform experiments that vary the retransmission rate $r$ and the resurrection rate $\gamma$. For each dataset, we run simulations for several values of $r$ and $\gamma$, and record $C_{``\infty"}$, the number of carriers left after a "long" time ($1,000$ simulation epochs, in our experiments). Recall that we defined this situation as quasi-steady-state, and we defined $C_{``\infty"}$ as the number of *residual carriers*.

Again, for each setting we run $100$ simulations, to obtain confidence intervals.

**Varying retransmission rate $r$:** Here we fix the death rate $\delta$ and the resurrection rate $\gamma$ both to $0.01$. Figure 5 shows the number $C_{``\infty"}$ of residual carriers versus the retransmission rate $r$, on all four datasets. The results of our dynamical system (Eqs. 1-3) were very close to that of the simulations, and are omitted for visual clarity. The dashed vertical line marks the "at-threshold" setting, that is, the value of $r$ that gives a survivability score of $s = 1$. As earlier, the small sizes of networks *INTEL* and *MIT* have higher variance.

We observe the following:

• *Below our threshold, the information dies out:* The number of carriers is very close to zero for all the datasets.

• *Above the threshold, the information survives:* Even after a "long" time, there is a significant population of nodes in the network that are alive and carry the information.

• *Effect of network size*: rTthe larger the network, the more accurately our theorem marks the onset of survivability. The results are good for *INTEL* and *MIT* ($N = 54$ and $N = 40$), very good for *GRID* ($N$=10,000) and perfect for *GNUTELLA* ($N$=62,000),

In conclusion, our threshold condition is very accurate.

**Varying resurrection rate $\gamma$:** Here we vary the resurrection rate $\gamma$, while keeping the the rest fixed ($r$=0.1, $\delta$=0.01). Figure 6 shows the results in an analogous fashion to Figure 5. The conclusions are identical as in Figure 5, providing additional evidence that our threshold condition is accurate.

### D. Insensitivity to initial conditions

So far we have considered the case where all nodes are initially in the 'Has Info' state, i.e. all nodes are carriers (infected). Next we show that our results do not change as we vary the number of initial carriers $\bar{C}(t = 0)$.

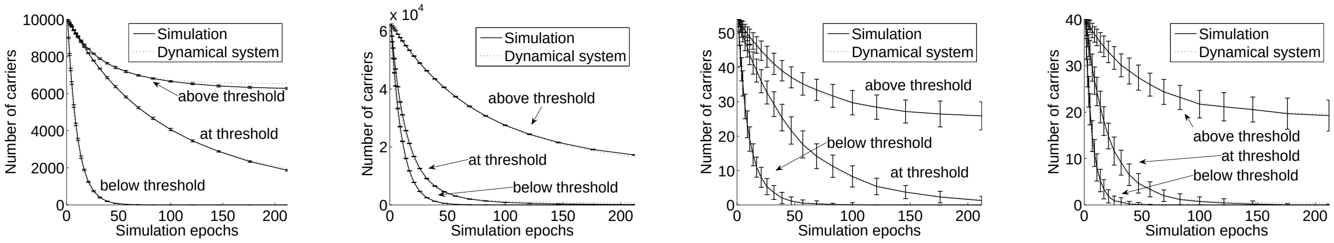
Figure 7 shows examples of the network being below-, at- and above- the threshold. We run the experiment on the *GNUTELLA* network with $N$ = 62,000 nodes and $E$ = 295,000 edges. We vary the number of initially infected nodes in the range (1,000, 5,000, 10,000, 20,000, 40,000). Notice that the behavior is independent of the starting conditions. In Figure 7(a), below the threshold, the information dies out exponentially fast. Not surprisingly, with fewer initial carriers $\bar{C}(t = 0)$, the information becomes extinct even faster. Figure 7(b) shows the number of carriers over time, when we are at threshold. Now, information is dying out much slower (polynomially fast).

In Figures 7(c, d), we are above the threshold. Notice that the information now survives. Moreover, all curves converge to the same expected number of carriers $C_{``\infty"}$, *regardless* of the initial conditions. In Figure 7(c), notice that some curves move up, while some others move downwards, so that they all reach the same state. The curves in Figure 7(d) are qualitatively similar, with just a higher $\bar{C}(t)$ (thanks to the higher survivability score $s$).

## V. Conclusions

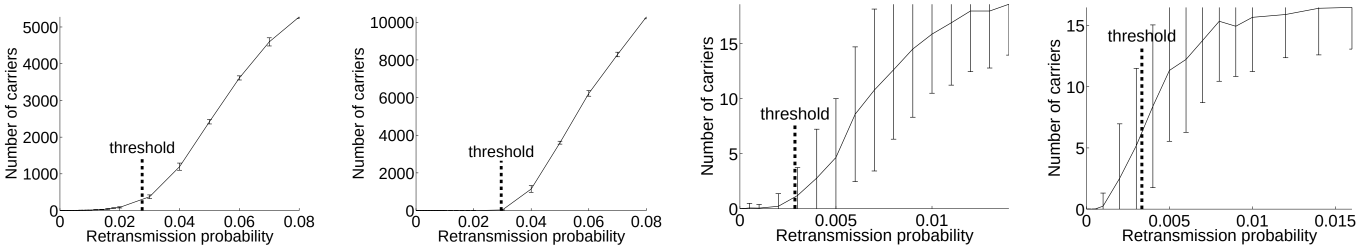
We formulated and studied the problem of the "information survival threshold", that is, the condition under which a datum transmitted from node to node will survive, despite node and link failures. Our contributions are the following:

- *Closed form formula*: We provide the first and only solution to this problem, with a simple formula that works for arbitrary network topologies, and arbitrary rates for retransmission, death and resurrection.



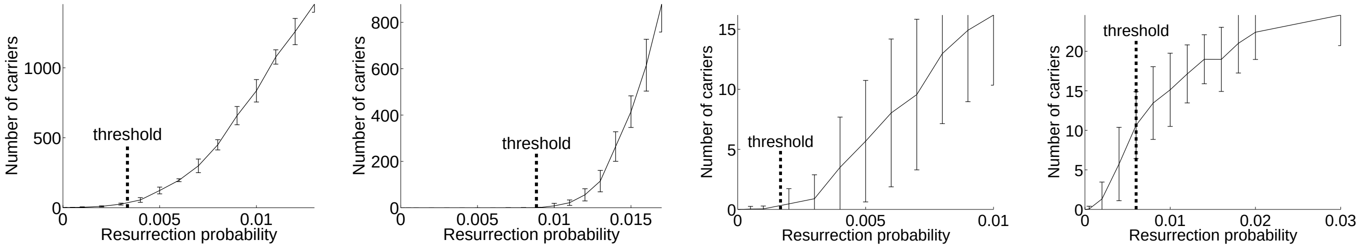
(a) *GRID* (b) *GNUTELLA* (c) *INTEL* (d) *MIT*

Fig. 4. *Number of carriers versus time (simulation epochs):* Our dynamical system (dotted lines) and simulation (solid lines). Confidence intervals show +/- one standard deviation. For each dataset we have three cases: below-, at-, and above-threshold, with parameter settings shown in Table II. Notice that (1) The dynamical system (dotted lines) is very accurate, being close to the simulation (solid lines), and (2) the number of carriers dies out very quickly below the threshold, while the information "survives" above the threshold.



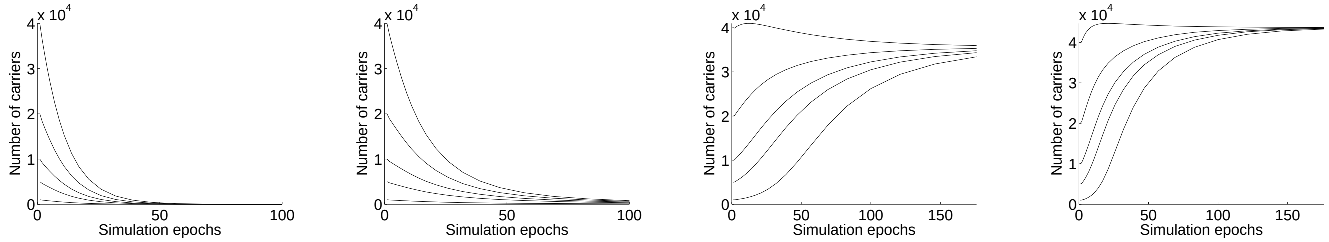
(a) *GRID* (b) *GNUTELLA* (c) *INTEL* (d) *MIT*

Fig. 5. *Number of "residual carriers" $C_{``\infty"}$, versus the retransmission probability:* The dashed vertical line marks our threshold ($s = 1$). The information dies out below our threshold, but survives above it.



(a) *GRID* (b) *GNUTELLA* (c) *INTEL* (d) *MIT*

Fig. 6. *Number of "residual carriers" $C_{``\infty"}$, versus the resurrection probability:* The dashed vertical line marks our threshold. Again, our threshold is very accurate.



(a) Below the threshold, $s$=0.912 (b) At the threshold, $s$=1.003 (c) Above the threshold, $s$=1.1 (d) Much above the threshold, $s$=1.2

Fig. 7. *Number of carriers $\bar{C}(t)$ versus time (simulation epochs): GNUTELLA* network, below, at and above the threshold, with initial number of carriers $\bar{C}(t = 0)$ varying in (1,000, 5,000, 10,000, 20,000, 40,000). The extinction/survival behavior is the same, regardless of the initial conditions.

- *Experiments on real data* Through extensive experiments we show that our analysis is extremely accurate, with typical relative error about $1\%$.
- Several additional observations: (a) the final state does not depend on the initial conditions, and (b) our analysis includes the well known SIS infection model as a special case.

From a practical system design point of view, we avoid ("fast") extinction if we arrange the network topology and the network parameters (retransmission-, death-, and resurrection-rates) so that we satisfy our condition ($s = |\lambda_{1,S}| \geq 1$). And conversely, if we want to guarantee fast extinction (say, for a computer virus, or an illegal copy of an MP3 song), we should shoot for the reverse condition.

Future work could focus on optimization problems, where our result provides a valuable stepping stone. A typical target question would be *what is the cheapest (least energy) network that can sustain a 'datum'?* One could also study mechanisms by which nodes in a dynamic high-churn network could *determine* the current threshold, and act accordingly, like, e.g., stop retransmitting to save energy.

## References

[1] L. Allen and A. Burgin. Comparison of deterministic and stochastic sis and sir models in discrete time. *Journal of Mathematical Bioscience*, 163(1):1–33, 2000.

[2] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Hafner Press, 2nd edition, 1975.

[3] S. Bellovin, B. Cheswick, and A. D. Keromytis. Worm propagation strategies in an IPv6 Internet. *;LOGIN*, 31(1), 2006.

[4] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, , and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2):41–88, 1999.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D.Shah. Gossip and mixing times of random walks on random graphs. `www.stanford.edu/~boyd/reports/gossip_opt.pdf`.

[6] P. Bremaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues*. Springer, New York, 1999.

[7] M. Burns, A. George, and B. Wallace. Simulative performance analysis of gossip failure detection for scalable distributed systems. *Cluster Computing*, 2(2):207–217, 1999.

[8] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85, 2000.

[9] D. Chakrabarti. Tools for large graph mining. Technical Report CMU-CALD-05-107, Carnegie Mellon University, 2005.

[10] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. 2002.

[11] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *ICDE*, 2004.

[12] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinchart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *PODC*, 1987.

[13] R. Durrett and X.-F. Liu. The contact process on a finite set. *The Annals of Probability*, 16(3):1158–1173, 1988.

[14] R. Durrett and R. H. Schonmann. The contact process on a finite set II. *The Annals of Probability*, 16(4):1570–1583, 1988.

[15] R. Durrett, R. H. Schonmann, and N. I. Tanaka. The contact process on a finite set III: The critical case. *The Annals of Probability*, 17(4):1303–1321, 1989.

[16] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical Report IRB-TR-02-003, Intel Research, 2002.

[17] A. Ganesh, D. Gunawardena, P. Key, L. Massoulie, and J. Scott. Efficient quarantining of scanning worms: Optimal detection and co-ordination. 2006.

[18] A. Ganesh, L. Massoulié, and D. Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM*, 2005.

[19] M. Garetto, W. Gong, and D. Towsley. Modeling malware spreading dynamics. In *INFOCOM*, 2003.

[20] M. Garofalakis and A. Kumar. Deterministic wavelet thresholding for maximum-error metrics. In *PODS*, 2004.

[21] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

[22] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse. Kelips: Building an efficient and stable p2p dht through increased memory and background overhead. In *Proceedings of the International Workshop on Peer-to-Peer Systems*, 2003.

[23] I. Gupta, A.-M. Kermarrec, and A. J. Ganesh. Efficient epidemic-style protocols for reliable and scalable multicast. In *SRDS*, 2002.

[24] T. E. Harris. Contact interactions on a lattice. *Annals of Probability*, 2:969–988, 1974.

[25] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, 1974.

[26] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *SenSys*, pages 134–147, 2004.

[27] http://db.lcs.mit.edu/labdata/labdata.html.

[28] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Symposium on Foundations of Computer Science (FOCS)*, 2003.

[29] D. Kempe and J. Kleinberg. Protocols and impossibility results for gossip-based communication mechanisms. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 2002.

[30] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. In *ACM Symposium on the Theory of Computing (STOC)*, 2001.

[31] J. O. Kephart and S. R. White. Directed-graph epidemiological models of computer viruses. In *IEEE Symposium on Research in Security and Privacy*, 1991.

[32] J. Kim, S. Radhakrishnan, and S. K. Dhall. Measurement and analysis of worm propagation on internet network topology. In *Proceedings of the Internation Conference on Communication in Computer Networks*, 2004.

[33] P. Levis, N. Patel, S. Shenker, and D. Culler. Trickle: A self-regulating mechanism for code propagation and maintenance in wireless networks. In *Proceedings of NSDI*, 2004.

[34] T. M. Liggett. *Interacting Particle Systems*. Springer Verlag, 1985.

[35] C. Lindemann and O. P. Waldhorst. Modeling epidemic information dissemination on mobile devices with finite buffers. In *SIGMETRICS*, 2005.

[36] J. Luo, P. T. Eugster, and J.-P. Hubaux. Route driven gossip: Probabilistic reliable multicast in ad hoc networks. In *IEEE INFOCOM*, 2003.

[37] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. An analysis of the sapphire/slammer worm, 2003.

[38] R. V. Renesse. Scalable and secure resource location. In *Hawaii International Conference on System Sciences*, 2000.

[39] R. V. Renesse, R. Minsky, and M. Hayden. A gossip-style failure detection service. In *Conference on Distributed Systems Platforms and Open Distributed Processing Middleware*, 1998.

[40] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a DHT. In *USENIX*, 2004.

[41] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.

[42] K. Sistla, A. George, R. Todd, and R. Tilak. Performance analysis of flat and layered gossip services for failure detection and consensus in heterogeneous cluster computing. In *Proceedings of the Heterogeneous Computing Workshop (HCW) at the International Parallel and Distributed Processing Symposium (IPDPS)*, 2001.

[43] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *USENIX Security Symposium*, 2002.

[44] S. Tanachaiwiwat and A. Helmy. VACCINE: War of the worms in wired and wireless networks. In *INFOCOM*, 2006.

[45] R. W. Thommes and M. J. Coates. Epidemiological modelling of peer-to-peer viruses and pollution. 2006.

[46] S.-C. Wang and S.-Y. Kuo. Communication strategies for heartbeat-style failure detectors in wireless ad hoc networks. In *DSN*, 2003.

[47] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos. Epidemic spreading in real networks: An eigenvalue viewpoint. In *SRDS*, 2003.

[48] Y. Wang and C. Wang. Modeling the effects of timing parameters on virus propagation. In *Proceedings of the ACM workshop on Rapid Malcode*, 2003.

[49] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *IFIP Networking*, 2006.

[50] S. Q. Zhuang, D. Geels, I. Stoica, and R. H. Katz. On failure detection algorithms in overlay networks. In *IEEE INFOCOM*, 2005.

[51] C. Zou, D. Towsley, and W. Gong. Email worm modeling and defense. Technical report, 2004.