# Network Forensics: Random Infection vs Spreading Epidemic

Chris Milling
The University of Texas at Austin
Austin, TX 78712, USA
cmilling@mail.utexas.edu

Constantine Caramanis
The University of Texas at Austin
Austin, TX 78712, USA
caramanis@mail.utexas.edu

Shie Mannor
Technion, Israel Institute of Technology
Haifa 32000, Israel
shie@ee.technion.ac.il

Sanjay Shakkottai
The University of Texas at Austin
Austin, TX 78712, USA
shakkott@mail.utexas.edu

## ABSTRACT

Computer (and human) networks have long had to contend with spreading viruses. Effectively controlling or curbing an outbreak requires understanding the dynamics of the spread. A virus that spreads by taking advantage of physical links or user-acquaintance links on a social network can grow explosively if it spreads beyond a critical radius. On the other hand, random infections (that do not take advantage of network structure) have very different propagation characteristics.

If too many machines (or humans) are infected, network structure becomes essentially irrelevant, and the different spreading modes appear identical. When can we distinguish between mechanics of infection? Further, how can this be done efficiently? This paper studies these two questions. We provide sufficient conditions for different graph topologies, for when it is possible to distinguish between a random model of infection and a spreading epidemic model, with probability of misclassification going to zero. We further provide efficient algorithms that are guaranteed to work in different regimes.

## Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Stochastic Processes

## Keywords

epidemic process, network inference

## 1. INTRODUCTION

The degree of interconnection in communication and social networks is unprecedented, and by now, well-documented. While interconnection speeds the spread of information and ideas — another exhaustively studied research topic — it inevitably contributes to the spread of viruses. This is true not only because of (what we might now call) old-fashioned contact networks, but increasingly true thanks to social networks, and the fact that many computer (and human) viruses exploit the relaxed filters we all employ when in virtual or physical contact with friends and acquaintances – neighbors on our social network. The prevalence of (thus far, apparently relatively harmless) Facebook spam [1] is but one testament to this fact. Yet, while many viruses can spread in various manners through social networks, it is not difficult with only minor sophistication, for a virus to disguise its path, namely, how it arrived at a particular machine, and where it spread from there. To make matters worse, analogous to the setting where not all sick people immediately go to a doctor for diagnosis, for a variety of reasons we may not even have access to the identity of all infected nodes, let alone be able to determine the spreading mechanism. Nevertheless, distinguishing between an infection that spreads through a particular network, and one that affects nodes without the help of that network, can be critical: it offers opportunities to react accordingly, including quarantining portions of the network, as well as possibly predicting the extent of the spread.

This motivates us to study this basic problem in social-network-forensics, in the most dire setting: suppose that at a single snapshot in time, we are informed that a given subset of nodes has a particular virus. Initial sickness times are not available, nor are we able to observe the evolution of the sick-reporting process. Given (for now) complete knowledge of the network topology, the problem is to determine if the virus is an epidemic, spreading through the network, or if nodes have become infected via an independent infection mechanism that is external to the network being considered, and not propagated through the edges of the graph. These issues apply not only to "sickness" spreading in human/online social networks, but more generally as well. As a concrete example of these two different modes of "sickness", consider a computer network that undergoes cascaded failures due to to virus/worm propagation (the epidemic) vs. random failures due to misconfiguration whose stochastic behavior is external to the network itself (independent in-

*SIGMETRICS'12,* June 11–15, 2012, London, England, UK.

fections). When a small subset of nodes report failure, the objective is to determine which of these two modes has occurred.

We study this problem in four regimes: for a d-dimensional grid, for uniformly branching tree, and critically and also highly connected Erdös-Renyi graphs. Grids are in a sense local connectivity models, lacking long-range edges. They model contact networks that are very correlated with geography. They are characterized by slow spreading and small neighborhood growth. Trees model explosive (exponential) neighborhood growth. Also, while contact networks are rarely trees, they are often (locally) tree-like. Erdös-Renyi graphs are on the opposite extreme to grid graphs: there is no notion of near and far when it comes to an edge, as all edges are equally likely. They model contact networks with large neighborhoods, and small diameter.

As is to be expected, if too many nodes are infected, then the effect of network structure is washed away: it is effectively impossible to distinguish between a random infection model independent of the network, and a network-spread epidemic. Also, from a practical perspective, we are interested in detecting if there is an epidemic at an early-stage where only "few" nodes are infected, so that appropriate intervention strategies can be deployed to quell the epidemic. An interesting general finding is that the results are quite delicate, depending both on the topology of the graph, and the level of infection. What is meant by "too many infected nodes" crucially depends on both the topology of the network, as well as the algorithm used. Thus using carefully designed algorithms with performance guarantees in terms of their detection capability, is critical.

The intuition behind distinguishing the infection mechanism is simple: if the sick nodes are uniformly spread out on the network, a random infection is likely at work, while if they are "clustered" in some appropriate sense, then it is more likely that we have an infection on our hands. As we see, not all measures of "clustering" are equally powerful. Thus, finding efficiently computable and maximally discriminating measures of clustering is, in a sense, the heart of the problem. The main contribution of this paper is to provide the analysis of this problem for several different regimes, and to provide two efficiently implementable algorithms that each compute a particular measure of "spread." We call them the "Ball Algorithm" and the "Tree Algorithm" (they are described explicitly below). The Ball Algorithm considers the smallest ball that contains all the reporting sick nodes, and the Tree Algorithm considers the smallest tree connecting all reporting sick nodes (the Steiner tree). Both algorithms then declare "infection" or "random sickness" based on a threshold test. We analyze both algorithms, showing analytically and empirically where each is strongest. Figure 1 shows examples of this problem on a grid and an Erdös-Renyi graph.

## Related Work

The infection model we consider here is the susceptible-infected (SI) model [3]. Most of the work on this model has focused on the analytic side, characterizing the spread of the infection in different settings, e.g., for graphs with multiple mixing distances (that is, local and global spreading) [4], and where the infected nodes are mobile [5]. There are other approaches to modeling infection, and while interesting to extend the current ideas and analysis there, we do not consider these in the present work.

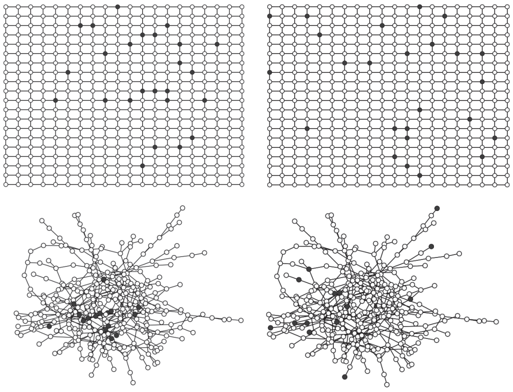

**Figure 1: This figure shows infection (left) vs random sickness (right) on the grid and the Erdös-Renyi graph. The figures have been generated using NetworkX [2].**

On the inference side, work in [6] provides a Bayesian inference approach for estimating the transmission rates of the infection. Alternatively, one can use MCMC methods to estimate the model parameters [7], [8]. A similar problem is considered in [9, 10], where, given a set of infected nodes, one seeks to determine which node is most likely to be the original source of the infection. These results typically involve approximation, at least for general graphs, due to the difficulty of exact inference for infections.

Several of our results here are related to first-passage percolation [11]. In the first-passage percolation basic formulation, there is a (lattice) graph of infinite size. For each edge, an independent random variable is generated that represents the time taken to traverse that edge. Some node is denoted the source, and the time taken to reach another node is the minimum of the total time to traverse a path over all paths between the source and that destination. This is equivalent to an infection traveling through the network as considered here. Work has been done to analyze interesting properties of this percolation, such as the shape of the infection and the rate at which it spreads. In particular, there are strong results on trees [12] and lattices [11] which we leverage.

## Outline of the Paper and Notation

The remainder of the paper is organized as follows. In Section 2, we precisely define the statement of the problem, and the two infection models which we later attempt to distinguish. We also give the Ball and Tree Algorithms whose performance we analyze in the sequel. Section 3 considers infections and node sickness in a grid topology. Section 4 then analyzes trees. Section 5 considers Erdös-Renyi graphs, where we find more complex behavior exhibited. Finally, we numerically illustrate our results and the performance of our algorithms on different graph topologies, in Section 6. Figure 1 shows examples of the infection on these graphs.

## 2. PROBLEM STATEMENT

This paper considers the spread of a virus or an infection among interconnected agents, that might represent computers, or perhaps humans. We model the interconnections via edges in a graph, with an edge representing interaction between two nodes. This could be physical interaction as in a contact model, or it might denote an acquaintance relationship as in a social network. The setting we have in mind is when the interaction represented by these nodes has the potential to spread certain types of viruses. In contact networks this might be due to direct infection, while in a social network this might be because the virus exploits the implicit trust represented by links (e.g., click-through more likely for a malicious link sent from a friend's account than from a stranger's).

We note that while this paper uses the language of *infection* and *virus*, our results could equally well be understood in the context of *information* or some other abstract quality (e.g., owns an iPhone), where the mechanism of spread or acquisition is of interest (e.g., television commercials, or word-of-mouth). While we do not consider weighted graphs here, the related questions are clearly relevant, and are natural extensions of the work we present.

## Node Sickness Models

As discussed, we assume there are two possible causes of the sickness: *Random sickness*, where the sickness spreads randomly and uniformly over the network and in particular the network plays no role in spreading the sickness; and *Infectious spread*, where the sickness is caused through a contagion that spreads through the network, with one node infecting its neighbors with some probability. We wish to distinguish between these two modes of sickness, *when only a small sub-collection of nodes report sick.* More precisely, we have:

> *Random Sickness*: Each node becomes sick with probability $\hat{q}_1$, independently of all other nodes. At a fixed time $t$ each sick node reports sickness to a central authority with probability $\hat{q}_2$, again independently of all other nodes. Thus on average, a fraction $\hat{q}$ of the network reports sick, where we let $\hat{q} = \hat{q}_1 \cdot \hat{q}_2$.
>
> *Infectious Spread*: At time 0, a randomly selected node in the network becomes infected. When a node becomes infected, we start a clock for each outgoing edge attached to an adjacent node that is not already infected, with expiration time exponentially distributed with unit mean (independent of any other event). Upon expiration of a edge's clock, the corresponding node becomes infected (if by then it is not already infected through some other neighbor), and in-turn begins infecting its neighbors. Alternatively, the infection can be said to travel at rate 1 along the edges of the graph. In this way, the infection spreads through the graph for time $t$. At this time, as in the network-free sickness model, sick nodes report the sickness independently with probability $q$. For $q = 1$, the reporting sick nodes form a connected component, making the detection problem trivial. For small values of $q < 1$, however, reporting nodes need not be adjacent, and can in fact be far from other reporting sick nodes.

Naturally, the problem is of most interest when the probabilities of sickness, reporting, and also time that the sicknesses are collectively reported/discovered in the spreading model, are such that the expected numbers of sick nodes and reporting sick nodes are equal under both models, as this is when the discrimination problem is most challenging. We refer to "infected nodes," denoted by $I$, when the model is an infection, and "sick nodes," denoted by $S$, when the model is a random sickness. We use $I_r$ and $S_r$ to denote the subset of reporting sick nodes in each setting. The case of most interest is thus when $\mathbb{E}[S_r] = \mathbb{E}[I_r]$. When only a few (say, two or three) nodes report sick, distinguishing between the two models is easy. On the other extreme, when all or nearly all of the network is infected, distinguishing the two modes of sickness is impossible. Thus one way to talk about the power of a given algorithm on a given graph topology, is according to how many nodes can report sick and still have successful discrimination between the two models. Equivalently, we talk about the "time" $t$, and normalize such that at this fixed time $t$, $\mathbb{E}[S_t] = \mathbb{E}[I_t]$. While we explicitly use the knowledge of $t$ in the algorithms and results, we note that it is possible to *estimate* this parameter $t$ using only the number of sick people reporting, and knowledge of spreading and reporting rates. It turns out, this does not materially change the convergence results up to poly-log factors (i.e., the estimates appropriately concentrates to the true value in the large $n$ setting). However, in this paper, we simply assume that $t$ is known for ease of discussion.

Accordingly, in our analysis below, we seek to provide bounds on precisely this: what is the maximum number of reporting sick nodes we can have, while still maintaining correct detection with probability one (asymptotically). We assume that the *a priori* probabilities of each type of sickness are equal, and define the event *error* as incorrectly identifying the type of sickness. Then using this notation, we would like *upper* bounds on $t$, or equivalently $\mathbb{E}[S_r]$ (which scales with graph size), for which we can guarantee that if the sickness is discovered before time $t$, or before $\mathbb{E}[S_r]$ nodes are infected, then we can discriminate between the two spreading mechanisms, i.e., $P(error) \to 0$ as $n \to \infty$. We note again that we consider the problem where we have no knowledge of the evolution of the sick-reporting process. This is an interesting direction, which would surely enable *more powerful* algorithms that can exploit this additional time-evolution information.

## The Ball and Tree Algorithms

We consider two algorithms for solving this problem. We compare their error rates analytically in Sections 3, 4 and 5, and then empirically in Section 6. These algorithms examine the set of reporting sick nodes and calculate a 'score' that rates in their respective ways how clustered the sick nodes are. They take as a parameter a threshold $m$ and if the score is no more than that threshold, they report that the sick nodes appear to come from an infection. The threshold parameter $m$ is topology-dependent, and we provide values for it in each relevant section below. We denote the distance between nodes $u$ and $v$ by $dist(u, v)$.

We term the first algorithm the 'Ball Algorithm'. We find the center of the reporting sick nodes, and then compute the radius of the ball containing all reporting sick nodes. If this distance is no more than $m$, we call the sickness an infection because it appears clustered. Otherwise, we call it a random sickness.

The second algorithm is called the 'Tree Algorithm.' We find a tree with the smallest number of nodes that connects each reporting node – this is called the Steiner tree [13]. If the number of nodes in the tree is less than the (again topology-dependent) threshold $m$, we call the sickness an

**Algorithm 1** Ball Algorithm

**Input:** Set of reporting sick nodes $S$; Threshold $m$
**Output:** INFECTION or RANDOM

```
$k \leftarrow \infty$
for all $v \in V$ do
  $d \leftarrow 0$
  for all $u \in S$ do
    if $dist(u, v) > d$ then
      $d \leftarrow dist(u, v)$
    end if
  end for
  if $d < k$ then
    $k \leftarrow d$
  end if
end for
if $k \leq m$ then
  return INFECTION
else
  return RANDOM
end if
```

infection. Otherwise, we call it a random sickness. Finding the minimum Steiner tree is an NP-hard problem, though there are efficient algorithms that give approximate solutions, guaranteeing no more than twice the optimum number of nodes or better [14].

We analyze this inference problem and in particular the performance of our two algorithms, on three types of graphs. First, we consider an infection on a $d$-dimensional grid. In this case, both our algorithms are able to (asymptotically) eliminate Type I and Type II error, for up to a constant fraction of sick nodes, even when only a logarithmic fraction report sick. Orderwise, this is the best any algorithm (regardless of computational complexity) can hope to achieve. Our empirical results verify this performance, and also show that the Ball Algorithm outperforms the Tree Algorithm on the grid.

**Algorithm 2** Tree Algorithm

**Input:** Set of reporting sick nodes $S$; Threshold $m$
**Output:** INFECTION or RANDOM

```
$T = SteinerTree(G, S)$
$k = \text{card}(T)$
if $k \leq m$ then
  return INFECTION
else
  return RANDOM
end if
```

Next we consider tree graphs. Here we show that the Tree Algorithm can correctly discriminate between infections and random sickness for larger numbers of reporting sick nodes than the Ball Algorithm is able to handle. Finally, we analyze Erdös-Renyi graphs under two different connectivity regimes: a low-connectivity with edge probability close to the regime when the giant component emerges; and a high connectivity regime the produces densely connected graphs. Again, we show that each algorithm can identify an infection with probabilities of error that decay to 0 as the network size goes to infinity, for appropriate ranges of parameters. Not surprisingly, the more densely connected, the more difficult it becomes to obtain a good measure of 'clustering.' Consequently, in these latter regimes, we find that one needs to intercept the sickness much earlier, i.e., with many fewer reporting sick nodes, in order to hope to accurately discriminate between the two potential sickness mechanisms. In the Erdös-Renyi setting, we are unable to find direct analytic results to compare our two algorithms. However, in Section 6 we evaluate them empirically and find that the Ball Algorithm tends to perform better, despite its relative algorithmic simplicity.

## 3. MULTIDIMENSIONAL GRIDS

The phenomenon of a sickness that is "going around" the office, the neighborhood, the school, is an instance of a contact network that is geographic – distance on the graph is closely related to geographic distance, and thus there are no 'long hops' in the graph. A canonical graph from this family is the $d$-dimensional grid, and we consider this first. Thus, let the graph $G$ be such a grid network with $n$ nodes and side length $n^{1/d}$. To avoid edge effects, we let the opposite edges of the grid connect, so that the graph forms a torus. Now due to the symmetrical structure of the grid, the initial source of an infection does not change the behavior of the infection.

Now we establish that both our algorithms work very well on this graph, even in the case of very low reporting rates. First we consider the Ball Algorithm. It turns out this performs very well on this very structured graph, in part because it matches well with the expected shape of an infection on a grid. The Tree Algorithm can also be shown to work on this graph for smaller infections, though this proof has been omitted for brevity and because it is heavily outperformed by the Ball Algorithm in this setting.

The next theorem gives conditions on the performance of the Ball algorithm, when time $t$ has elapsed (recall that this is equivalent to fixing the expected size of the infection, and we assume for now that this is known). The number of reporting nodes can be arbitrary, but must be above $\log n$. Note that this requirement, along with the time $t$, implicitly constrains the underlying parameters of the problem setup, namely $q$. We use $\mu$ to denote the expected rate that an infection travels along an axis on the grid, which is only a function of the dimension of the graph, since we assume the spreading rate to be normalized. We have the following.

THEOREM 1. *Consider the Ball algorithm (Algorithm 1) with threshold $m = 1.1d\mu t$. Suppose that the expected number of reporting nodes scales at least as $\log n$. Then there exists constant $C_1$ such that for sufficiently large $n$, if the expected number of infected nodes is less than $C_1 n$,*

$$P(error) \to 0.$$

*In other words, an infection can be identified with probability approaching 1 as $n$ tends to infinity.*

To prove this theorem, we use a previous result that characterizes the spread of an infection. Since we model the time it takes the infection to traverse an edge as an independent exponentially distributed random variable, the time a node is infected is the minimum sum of these random variables over all paths between the infection origin and that node.

This simply phrases the infection process in terms of first-passage percolation on this graph. This allows us to use a result characterizing the 'shape' of an infection on this graph (see [11]). Let $I(t)$ be the set of infected nodes at time $t$. Imagining the graph as the integer lattice embedded in $\mathbb{R}^d$ with the infection starting at the origin, let us put a small $\ell^\infty$-ball around each infected node. This allows us to simply state inner and outer bounds for the shape of the infection. To this end, define this expanded set as $B(t) = I(t) + [-1/2, 1/2]^d$.

Lemma 1 ([11]). *There exists a set $B_0$ and constants $C_1$ to $C_5$ such that for $x \leq \sqrt{t}$,*

$$P\{(B(t)/t \subset (1 + x/\sqrt{t})B_0)\} \geq 1 - C_1 t^{2d} e^{-C_2 x}$$

*and*

$$\begin{aligned} &P\{(1 - C_3 t^{-1/(2d+4)}(\log t)^{1/(d+2)})B_0 \subset B(t)/t\} \\ &\geq 1 - C_4 t^d \exp\left(-C_5 t^{(d+1)/(2d+4)}(\log t)^{1/(d+2)}\right). \end{aligned}$$

That is, the shape of the infected set $B(t)$ can be well-approximated by the region $tB_0$.

Moreover, one can show that this set $B_0$ is regular in that it contains an $\ell^1$-ball and is contained in an $\ell^\infty$ ball: $\{x : \|x\|_1 \leq \mu\} \subset B_0 \subset [-\mu, \mu]^d$, where $\mu \triangleq \sup_x\{(x, 0, ..., 0) \in B_0\}$, effectively the rate the infection spreads along an axis [11]. Note that $\mu$ does not depend on the *realization*. This result says that the expected shape of the infection is "nearly" a ball, in the sense described above. Therefore it is not surprising that our Ball Algorithm should do well.

Now we turn back to the finite grid with side length $n^{1/d}$ to present the proof our theorem.

Proof of Theorem 1. To prove this theorem, we prove the following more general statement. Let $m$ be the threshold for the Ball Algorithm and suppose $(2m/d + 1) < n^{1/d}$. If for some $\epsilon > 0$,

$$t < \frac{m}{d\mu(1+\epsilon)},$$

the Type II error probability decreases to 0 as $t$, $m$, and $n$ increase. In addition, the Type I error probability also decreases to 0 in the limit if

$$t^d q \left(\frac{n^{1/d} - 2m/d - 1}{n^{1/d}}\right) = \omega(\log n).$$

We begin with the Type II error probability, which we denote by $E_{II}$: the probability we mistake an infection process for a random sickness. As long as $m$ is chosen as in the statement of the theorem, we are guaranteed that if the sickness is in fact from an infection, then using the above lemma, the spread of the infection is limited to the sub-grid $[-m/d, m/d]^d$ with high probability, where the origin is set to be the original infected node. Consequently, all nodes must be within $m$ steps of the origin since the grid is $d$-dimensional. That is, we have

$$\begin{aligned} E_{II} &< 1 - P\{B(t) \subset [-m/d, m/d]^d\} \\ &< C_1 t^{2d} e^{-C_2 t^{-1/2}(m/(d\mu) - t)}, \end{aligned}$$

from Lemma 1, where we use $x = \min(t^{-1/2}(m/(d\mu) - t), t^{1/2})$. Therefore, given $\epsilon > 0$, $t < \frac{m}{d\mu(1+\epsilon)}$, indeed the error goes to 0 as $t$ and $n$ increase.

Next, we consider Type I error, $E_I$: the probability we mistake a random sickness for an infection process. This happens if all the reporting sick nodes happen to fall inside the ball of radius $m/d$. Recall that our problem is only of interest if the two processes yield roughly the same number of sick nodes reporting. We can get a lower bound on this number for the infection process (and hence for the random sickness process) this time using the inner bound on $B_0$. For the infection process, the second part of Lemma 1 asserts that the infected region contains all nodes within the $l1$-ball of radius $w = (1 - C_3 t^{-1/(2d+4)}(\log t)^{1/(d+2)})\mu t$ with probability at least $1 - P_1$, where

$$P_1 = C_4 t^d \exp\left(-C_5 t^{(d+1)/(2d+4)}(\log t)^{1/(d+2)}\right).$$

Therefore at least $2\lfloor w \rfloor^d$ nodes will be sick with that probability, and hence there will be on average, at least $2q\lfloor w \rfloor^d$ sick nodes reporting. What is the probability that the random sickness model with (at least) this many sick nodes will have all reporting nodes inside the sub grid $[-m/d, m/d]^d$? There are $L = (2m/d + 1)^d$ nodes in that region. Evidently, any given sick node satisfies that property with probability $L/n$, so they all satisfy it with probability at most $(L/n)^{2qw^d}$. Note that any dependence between sick nodes only reduces the probability. After this, we use a union bound to find that the probability no such region contains all sick nodes is at most $P_2 = n(L/n)^{2qw^d}$.

Putting it all together, we have,

$$\begin{aligned} E_I &< 1 - (1 - P_1)(1 - P_2) < P_1 + P_2 \\ &< C_4 t^d \exp\left(-C_5 t^{(d+1)/(2d+4)}(\log t)^{1/(d+2)}\right) \\ &+ n\left(\left(\frac{2m/d + 1}{n^{1/d}}\right)^d\right)^{2qw^d}. \end{aligned}$$

and

$$2w^d \geq 2\mu^d t^d (1 - dC_3 t^{-1/(2d+4)}(\log t)^{1/(d+2)}).$$

Note that $P_2$ dominates as $n$ increases. We want to find the regime when this probability tends to 0. That is, we want

$$\begin{aligned} &n \exp(2\mu^d t^d q d \ln\left(1 - \frac{n^{1/d} - 2m/d - 1}{n^{1/d}}\right) \\ &(1 - dC_3 t^{-1/(2d+4)}(\log t)^{1/(d+2)})) \to 0. \end{aligned}$$

Using a Taylor expansion and some simplification, we find a sufficient condition for this is that

$$t^d q \left(\frac{n^{1/d} - 2m/d - 1}{n^{1/d}}\right) = \omega(\log n).$$

This completes the proof of the general statement. In addition, the Type I error can be shown to dominate in the range of interest. Theorem 1 follows immediately using the threshold provided.

□

## 4. TREES

We now turn to the problem on tree graphs. Trees have different (exponential) spreading rates from grids, and hence manifest fundamentally different behavior. Indeed, while simple, tree graphs convey the key conceptual point of this section: the difficulty of distinguishing an epidemic from a random sickness on graphs where the infection spreads quickly. In addition, while the results do not immediately carry over, the behavior on a tree provides an intuition for the behavior of an infection on an Erdös-Renyi graph, which we cover in the next section.

Thus, let $G$ be a balanced tree with $n$ nodes, constant branching ratio $c$, and a single root node $a$. In the case of an infection, instead of choosing a node at random to be the original source of the infection, we always choose the root of the tree. This is the most interesting case, since otherwise a constant fraction of the nodes are very far from the infection source and bottlenecked by the root node. Also, this precisely models the scenario for locally tree-like graphs, such as Erdös-Renyi graphs.

First we examine the performance of the Ball Algorithm on this graph. Again recall the meaning of $t$: it is the time at which the sicknesses are reported, and also a proxy for the expected number of infected nodes.

THEOREM 2. *Consider the Ball algorithm (Algorithm 1). Suppose that the expected number of reporting nodes scales at least as* $\log n$. *Then there exist constants* $b$, $\beta$ *such that if the threshold* $m = 1.1bt$ *and the expected number of infected nodes is less than* $n^{\beta}$*,*

$$P(error) \to 0.$$

*In other words, an infection can be identified with probability approaching* $1$ *as* $n$ *tends to infinity.*

PROOF. To prove this theorem, we prove the following more general statement:

For some constant $\beta < 1$, if $qE[I] = \omega(1)$ and $E[I] < n^{beta}$, then the Type I error probability tends to 0. Next, there exists a constant $b$ such that if $b_0 > b$ and the threshold $m > b_0 t$ for all $n$, then the Type II error probability converges to 0 asymptotically, as the tree size scales.

The Type II error bound follows from results in first passage percolation [15]. In particular, one can compute the fastest-sustainable transit rate. This quantity is basically the time from the root to the leaves, normalized for depth, as the size of the tree scales. Formally (again, see [15] for details), let us consider a limiting process of trees whose size grows to infinity, with $\Gamma_n$ denoting the balanced tree on $n$ nodes, and $\delta(\Gamma_n)$ denoting the set of paths from the root to the leaves, and for a node $v \in p$ for some path $p \in \delta(\Gamma_n)$, let $X_v$ denote the time it takes the infection to reach node $v$. Then the *fastest-sustainable transit rate* is defined as: $\lim_n \inf_{p\in\delta(\Gamma_n)} \limsup_{v\in p} \frac{X_v}{\text{depth}(v)}$. Basic results [15] show that this quantity exists, and thus shows that the rate at which an infection travels, defined as the maximum distance of the infection from the root over time, converges to a constant $b$ that depends on the branching ratio. The probability that an infection travels at a faster rate converges to 0 in the size of the tree. This establishes the Type II result.

The Type I error result follows simply as well. Given the branching ratio, $c$, there are $\frac{c^{m+1}-1}{c-1}$ nodes within a distance $m$ from the root. Again letting $S_r$ denote the number of reporting sick nodes, the probability of a Type I error is (approximately) $(\frac{c^m}{n})^{S_r}$ – the probability that the randomly sick nodes are closer than the threshold $m$ to the root. Then if $c^m$ is $o(n)$, it is sufficient that the probability that $S_r = 0$ goes to 0. This occurs if the expected number of reporting sick nodes is $\omega(1)$. That is, we need $qE[I] = qe^{(c-1)t} = \omega(1)$, calculating $E[I]$ with a simple differential equation. Alternatively, if $c^m = \alpha n$ for some constant $\alpha < 1$, then we require $S_r$ to increase with $n$ with probability 1. The same condition as before is sufficient for this to be true. This completes the Type I result.

Using both these results, there is a choice of $m$ such that both error types become rare as long as $c^{b_0 t} < \alpha n$, so $c^t < (\alpha n)^{1/b_0}$. The theorem follows using a particular threshold. □

Next, we consider the Tree Algorithm on this graph. The threshold here depends on $E[I]$ instead of depending explicitly on $t$, but as discussed previously, these are essentially equivalent.

THEOREM 3. *Consider the Tree algorithm (Algorithm 2) with reporting probability* $q > \log\log n/\log n$ *and threshold* $m = E[I]\log\log n$. *Suppose that the expected number of reporting nodes scales at least as* $\log n$. *Then for any constant* $\alpha < 1$*, if the expected number of infected nodes scales as less than* $n^{\alpha}$*,*

$$P(error) \to 0.$$

*In other words, an infection can be identified with probability approaching* $1$ *as* $n$ *tends to infinity.*

PROOF. We prove the following generalization of the theorem: The Type I error probability converges to 0 for any choice of the threshold $m = o(qE[I]\log n)$ with $qE[I] = O(n^{\alpha})$ for some $\alpha < 1$. In addition, the Type II error probability converges to 0 if $m = \omega(E[I])$.

To prove the Type II error result (mistaking an infection for a random sickness), note that the size of the infection is $E[S] \le e^{(c-1)t}$. Since the Steiner tree containing the reporting nodes can be no larger than the infection itself, the Type II error converges to 0 as long as we use a threshold $m = \omega(E[I])$ from Markov's inequality. Next, we evaluate the Type I error probability (mistaking a random sickness for an infection). This requires estimating the size of the Steiner tree containing the reporting sick nodes. Suppose there is an $\alpha < 1$ such that $E[S_r] = O(n^{\alpha})$. Since the number of sick nodes increases with $n$, the probability that there are sick nodes on at least two subtrees of the root node goes to 1, hence the root of the tree is in the Steiner tree connecting the randomly sick nodes with high probability. Given this, we see that a node is in the Steiner tree if and only if it is infected or a node below it in the tree is infected. Let $N = S_r$. Since $E[S_r]$ is $\omega(1)$, $N$ is $\omega(1)$ with high probability. Choose the first level in the tree that has at least $N/c$ nodes. Then there are between $N/c$ and $N$ subtrees below that level. It is straightforward to show that each sick node in the tree has at least a 1/2 probability of being a leaf node since $c \ge 2$. Since at least $N$ nodes are sick, at least $N/4$ of the leaf nodes are sick and distributed independently among the at most $N$ subtrees. Therefore, the total number of subtrees with sick nodes at the bottom is at least $N/(8c)$. In addition, each leaf node in a separate subtree requires a path at least up to the aforementioned level in the Steiner tree. This gives us the following high probability bound on the Steiner tree size.

$$\begin{aligned}\text{Steiner Tree Size} &> \frac{N}{8c}(\log_c n - \log_c N)\\ &> N\frac{(1-\alpha)\log_c n}{8c}\\ &= S_r\frac{(1-\alpha)\log_c n}{8c}.\end{aligned}$$

For any $w = o(E[S_r])$, we know that $S_r > w$ with probability approaching 1, since $E[S_r] = E[I_r]$. Also, if $E[S_r] = O(n^\alpha)$, then $S_r = O(n^\alpha)$ with high probability. Therefore, if $m = o(qw \log_c n)$, which is equivalent to $m = o(E[S_r] \log n)$, the Type I error probability tends to 0. □

Note that the Ball Algorithm succeeds until the farthest infected node reaches the edge of the graph. At this point, the ball radius can increase no further, thus there is no hope of distinguishing an infection from a random sickness. Since this farthest point travels at a faster rate than the bulk of the infection, the Ball Algorithm can only work up to some time $\log_c n/b$. However, the Tree Algorithm can still correctly identify an infection with high probability nearly to the point where $\Theta(n)$ nodes are sick. This includes infection times close to $\log_c n$, the time it takes for every node to be infected. From this, we see that the Tree Algorithm works for a wider range of times compared to the Ball Algorithm. This is demonstrated by simulations in Section 6.

# 5. ERDÖS-RENYI GRAPHS

In this section, we consider Erdös-Renyi graphs. A notable difference in the topology of Erdös-Renyi graphs and grids is that the diameter of the former scales much more slowly (logarithmically) with graph size. That is, Erdös-Renyi graphs are more highly connected, in the sense that no two nodes are too far apart. These model contact graphs that are not tightly correlated to geographic proximity. For example, one might consider physical proximity forced by an event (a concert) attended by neighbors and non-neighbors alike, or a virtual network where edges are formed independent of geography.

We consider two connectivity regimes: the regime where the giant component first emerges, and each node has a constant expected number of edges, and then a much more highly connected regime, where the graph demonstrates different local properties, and discrimination between random sickness and infection is harder still.

## 5.1 Detection with Constant Average Degree

We first consider Erdös-Renyi graphs with constant average degree. Define the graph $G = G(n, p)$ to be the graph with $n$ nodes and for each pair of nodes, there is an edge between them with probability $p$. In the section above, we used $c$ to denote the branching ratio. We overload notation and use it again to measure the spread of the graph, but here as the expected degree: let $p = c/n$ with $c > 1$. In this regime, the graph is almost surely disconnected, but there is a giant component. Since this problem would be trivial on a disconnected graph, we limit both the infection and random sick nodes to the giant component. We show that unlike the case of trees, we are unable to distinguish infection from random sickness for close to a constant fraction of nodes. Instead, we consider infections that cover only $o(n)$ nodes. As is well-known (e.g., [16]) in this connectivity regime, the graph is locally tree-like, and hence tree-like in the infected region. This allows us to leverage some results from the previous section, although direct translation is not possible, particularly in the analysis of our second algorithm.

Again we note that in the next two theorems, the threshold depends on $t$ and $E[I]$, respectively. As discussed, these are essentially equivalent, and the choice amounts to ease of notation and exposition.

THEOREM 4. *Consider the Ball algorithm (Algorithm 1). Suppose that the expected number of reporting nodes scales at least as* $\log n$*. Then there exist constants* $b$, $\beta$ *such that if the threshold* $m = 1.1bt$ *and the expected number of infected nodes is less than* $n^\beta$*,*

$$P(error) \to 0.$$

*In other words, an infection can be identified with probability approaching* 1 *as* $n$ *tends to infinity.*

PROOF. The proof follows similar lines as in the previous section, so we omit most details. In particular, we show the following: Using a threshold $m < \frac{\log n}{2 \log c}$ and $qE[I] = \omega(1)$, the probability of a Type I error is at most $o(n^{-1})$. In addition, the probability of a Type II error converges to 0 as long as $m > bt$ for a constant $b$ specified in the proof.

To control the probability of a Type I error, we have to bound the probability that all randomly sick nodes are within a ball of radius $m$ on the graph. A sufficient condition for this is that all nodes are within distance $2m$ from a given sick node, or there are 0 nodes sick. The latter probability is simply $(1 - q)^n$ which decays exponentially. Also, with probability $1 - o(n^{-1})$, the number of nodes within a distance $2m$ from a given sick node is no more than $16m^3c^{2m} \log n$ [17]. Then the error probability in this case is at most $\left(1 - \frac{16m^3c^{2m} \log n}{n}\right)^n$. Then this decays exponentially as long as $c^{2m} = o(n)$, which occurs when $m < \frac{\log n}{2 \log c}$. Thus we find in total, the Type I error probability is $o(n^{-1})$.

We bound the probability of a Type II error again using the notion of the fastest sustainable transit rate from first-passage percolation [15]. As in Theorem 2, the constant $b$ comes from the calculation of the infection spreading rate, and the results follow similarly. □

The Tree Algorithm is more complex to analyze for this graph. The more delicate analysis comes from the challenge of bounding the size of the Steiner tree for the random sickness process, needed to control Type I error.

THEOREM 5. *Consider the Tree algorithm (Algorithm 2) with reporting probability* $q > \log \log n/\log n$ *and threshold* $m = E[I] \log \log n$*. Suppose that the expected number of reporting nodes scales at least as* $\log n$*. Then for any constant* $\alpha < 1/2$*, if the expected number of infected nodes scales as less than* $n^\alpha$*,*

$$P(error) \to 0.$$

*In other words, an infection can be identified with probability approaching* 1 *as* $n$ *tends to infinity.*

PROOF. We show the following more general statement: The Type II error probability decays to 0 if the threshold is chosen as $m = \omega(E[I])$ and $E[I] = o(n)$. The Type I error probability goes to 0 when $m < kqE[I]$ for some constant $k = o(\log(n/(qE[I])^2))$ and $qE[I] = o(\sqrt{n})$.

First, if the sickness is from an infection, the smallest tree connecting the reporting sick nodes must have size no more than the actual number of sick nodes. Hence, to bound the Type II error, it is sufficient to bound the probability the number of infected nodes is over a certain size. This probability decreases to 0 as long as $m$ is $\omega(E[I])$ when $E[I] = o(n)$. To see this, recall that in this regime, the graph looks locally tree-like. Consequently, we can bound the maximum number of infected nodes using bounds on the distance an infection can travel (e.g., see [15]). Again, Markov's inequality provides the exact error bound in the theorem statement.

To control Type I error probability, that a random sickness is mistaken for an infection, we must lower bound the size of the Steiner tree of a random sickness. Call this minimum value $r$. Again, let $S$ denote the number of sick nodes, and $\mathcal{S}$ the set of sick nodes. For $A \in \mathcal{S}$, let $d_A$ denote the distance from that node to the nearest other sick node. First we show that $\sum_{A \in \mathcal{S}} d_A \leq 2r$. Note that the bound is attained for some graphs, such as a star graph with the central node uninfected.

Consider the Steiner tree subgraph, and duplicate all edges on it. Since the degree of each node in the subgraph is even, there is a cycle that connects all these nodes. Naturally, the length of this cycle, which is twice the size of the Steiner tree, is larger than the length of the smallest cycle connecting all sick nodes. In addition, the length of this cycle is at least $\sum_{A \in \mathcal{S}} d_A$, since the distance from one sick node to the next sick node in the cycle is clearly no smaller than the distance from that sick node to the closest sick node. This establishes that $\sum_{A \in \mathcal{S}} d_A \leq 2r$.

Now we simply need to bound $d_A$. To do this, we need an understanding of the neighborhood sizes in a $G(n,p)$ graph. We provide an interesting way to visualize the distance distribution $f(d)$ – the distribution for the number of nodes at distance $d$. Then, using a CLT result, we provide a lower bound on the distance between one random node and the nearest sick node. To do this, we leverage concentration results for spreading processes on graphs. In order to not confuse this process with the sickness/infections we have been discussing, we follow standard terminology (e.g., [16]) and call a node "checked" if it has been reached by the infection, and "active" if it is checked but its neighbors are not all checked. Our process evolves as follows. At each time slot, we pick an active node at random, add its unchecked neighbors to the set of active nodes, and remove it from the active node set. If the set of active nodes ever becomes empty before the entire graph has been checked, we pick an unchecked node at random and make it active. This process continues until the whole graph is checked. Thus, at any time $t$ we have checked nodes $C_t$, active nodes $A_t$, and the remaining nodes $U_t$.

The following CLT result for this process is established in [16]. Define $\rho$ to be the solution to $\rho = \exp(c(\rho - 1))$. For any constant $\alpha$ such that $\alpha n$ is an integer, define $u_{\alpha n} = \text{card}(U_{\alpha n})$. Then, for $0 < \alpha < \rho$, $u_{\alpha n}$ converges to $ne^{-\alpha c}$. In particular,

$$(u_{\alpha n} - ne^{-\alpha c})/\sqrt{n} \xrightarrow{d} N(0, e^{-\alpha c} - c^{-2\alpha c}).$$

Now we illustrate how to use this result to determine the distance distribution for this graph. First, note that we can replace the random selection of an active node by always picking the closest node to the original node at each step in the process, and the CLT result remains unchanged. Thus, the process effectively performs a breadth first search of the graph. Eventually, we reach a state where the set $C_t$ of checked nodes contains all nodes at distance at most $d$ from the original node, $A$. Label the time this occurs $t_d$. Note that the number of nodes at distance $d+1$ is the set of active nodes, and $\text{card}(A_{t_d}) = n - u_{t_d} - t_d$. Then the time that we check all nodes within distance $d+1$ will be at $t_d + (n - u_{t_d} - t_d) = n - u_{t_d}$. This can be used to calculate the distance distribution. Letting time go to infinity and plotting on the graph with spacing $1/t$, we can think of $u_t$ as a curve. Using the above CLT result, the curve $y = n - u_t$ can be approximated by $y = ct$ for small $t$. That is, the distance distribution is close to that of a tree graph with the same branching ratio, $c$. Now we can calculate $f(d)$, the number of nodes at distance $d$. Let $\epsilon > 0$. It is simple to calculate by comparison of the two curves above, that with this branching process, $f(d+1) < (1+\epsilon)cf(d)$. Therefore, there are no more than $((1+\epsilon)c)^d$ nodes within distance $d$ when the process is close to its limit, which occurs with high probability provided that $d = \omega(1)$. Figure 2 illustrates how $f(d)$ is determined from these curves.

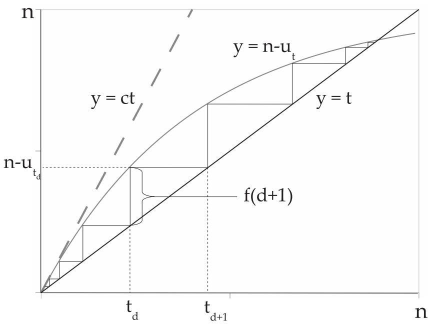

**Figure 2: Diagram illustrating how the limiting curve of this process can be used to determine the distance distribution.**

Now assume the number of sick nodes satisfies $S = o(\sqrt{n})$. Let $\epsilon > 0$ and $m = \epsilon n/S^2$. We let $k$ be a distance from an arbitrarily chosen sick node, $A$, within which there are at most $m$ nodes. Using the distance distribution calculation above, we find $k = o(\log(n/S^2))$ is sufficient for this condition. As the sick nodes are randomly selected, the probability that none of these are within a distance $k$ from $A$ is approximately $(1 - S/n)^m \to e^{-\epsilon/S} \to 1 - \epsilon/S$. Thus the distance to the closest sick node to $A$ is at least $k$, i.e., $d_A > k$, with high probability, and using a simple union bound, the same is true, simultaneously, for all sick nodes. Hence the Steiner tree joining the set of *reporting* sick nodes will be of size at least $r \geq (1/2)\sum d_A = (1/2)kqE[I]$, with probability decaying to zero. Again we have used the fact that $E[I] = E[S]$. Therefore, the Type I error probability tends to 0 as long as the threshold satisfies $m < kqE[I]$, for $k = o(\log(n/(qE[I])^2))$. Using this result, we find that the Tree Algorithm can succeed so long as $q\log(n/(qE[T])^2) = \omega(1)$. This is a complex condition, though the conditions given in the theorem are sufficient for it to be true. □

## 5.2 Detection on Dense Graphs

Now we consider the case of an Erdös-Renyi graph with a denser set of edges. Higher connectivity means the infection spreads faster, making it more difficult to distinguish between spreading mechanisms. The performance depends critically on the exact scaling regime. We consider the regime where there exists $d \in \mathbb{Z}$ and constants $\epsilon, h \in \mathbb{R}$ such that $\epsilon < n^{d-1}p^d < h$ holds for all $n$ as $n \to \infty$ (see also [18] for further discussion of this scaling regime and properties of these dense graphs). The next result bounds the size of the Steiner tree on a random collection of nodes, and is the key result for bounding of Type I error.

Lemma 2. *Suppose nodes are independently sick, with probability $n^{1/d}/n$, so that the expected number of reporting sick nodes is $qn^{1/d}$. Further suppose our graph is a $G(n,p)$ whose parameters satisfy $\epsilon < n^{d-1}p^d < h$ for $d > 4$. Let $Z$ be the size of the minimum Steiner tree connecting the reporting sick nodes. Also, let $m < (d-3)qn^{1/d}/2$ be the threshold for the Steiner tree size in the Tree Algorithm. Then $Z$ satisfies the following probabilistic limit: $\lim_{n\to\infty} Pr(Z < m) = 0$.*

Proof. Using precisely the same argument as above, we can lower-bound the size of the Steiner tree by $\sum d_A \le 2Z$, where the sum is over all reporting sick nodes, and as before, $d_A$ denotes the minimum distance from a reporting sick node $A$ to the nearest other reporting sick node. To lower bound the size of this sum, we rely on a result from [18] that shows that in this scaling regime, the asymptotic distribution of the distance between two random nodes is positive on only $d$ and $d+1$. That is, *almost all nodes* are either at distance $d$ or $d+1$ from any given source node $A$, and thus the function $f(d)$ defined previously, concentrates around $d$. To put this another way, let $F(d)$ be the probability that a random node is at distance more than $d$ from $A$. Then for any $\hat{d} > 1$, if $n^{\hat{d}-1}p^{\hat{d}} < h$, we have

$$\lim F_{\hat{d}} = \exp^{-n^{\hat{d}-1}p^{\hat{d}}}.$$

Now we condition on the number of sick nodes, $S$. Note $E[S] = n^{1/d}$ and the number of reporting sick nodes is just $q$ times this. We can compute the probability that the closest node is at distance more than $\hat{d}$ from $A$ simply as $F^I_{\hat{d}} \to \exp^{-(I/n)(np)^{\hat{d}}}$. Using our scaling regime, we know that $(\epsilon n)^{1/d} < np < (hn)^{1/d}$. To simplify notation, let $h' = h^{1/d}$. We have

$$\begin{aligned} F^I_{d-3} &\to 1 - I/n(np)^{d-3} \\ &> 1 - I/h'nn^{(d-3)/d}. \end{aligned}$$

Using a simple union bound, we find that the probability that some reporting sick node is within distance $d-3$ of another reporting sick node is at most $S^2/h'nn^{(d-3)/d}$. Since $S$ is a binomial random variable, it concentrates about its mean: for any $\epsilon' > 0, Pr((1-\epsilon')E[S] < S < (1+\epsilon')E[S]) \to 1$. Within this range, we find that $\sum d_A > (d-3)(1-\epsilon')E[S]$ with probability at least $1 - (1+\epsilon')^2 h' E[S]^2 n^{-3/d} > 1 - Cn^{-1/d}$ for some constant $C$. This converges to 1 for large enough $n$. Thus, we have shown the desired result. □

Now the probability of error calculations and hence the proof of correctness for the Tree Algorithm follows directly from the above.

Theorem 6. *For $G$ as above, suppose the expected number of reporting sick nodes in either model is $qn^{1/d}$. Then the Tree Algorithm with threshold $m$, the probability of a Type I error converges to $0$, as long as the threshold satisfies $m < (d-3)qn^{1/d}/2$. The probability of a Type II error upper bounded by $2/(d-3-\epsilon)$ as long as the threshold satisfies $m > (d-3-\epsilon)qn^{1/d}/2$, for any value of $\epsilon > 0$ such that $\epsilon + 3 < d$. This bound converges to $0$ as $d \to \infty$.*

Proof. Consider first the probability of a Type I error. This is the probability that a random sickness has a Steiner tree of size less than $m$. From Theorem 2, this probability converges to 0 if $E[I] = O(n^{1/d})$.

Second, consider the probability of a Type II error. As we have argued before, the size of this tree is no more than the total number of infected nodes, so it is sufficient to find the probability there are more than $m$ infected nodes. Using Markov's Inequality, this goes to zero when $m$ is an increasing factor greater than the mean number of infected nodes. □

# 6. SIMULATIONS

In this section we provide simulation-based evidence of the theoretical results of the previous sections. The simulations aim to demonstrate, in particular, two facts. First, the thresholds specified in the previous sections do actually work empirically, and as the graph size increases, the probability of both types of error decrease to zero. In addition, this provides insight into how quickly the probability of error decays. While our results include rate estimates given as part of the proof of correctness, we have not made an effort to optimize these in this work. Next, we seek to describe the relative performance of each algorithm, and show that it is as described above. Thus, we show that the Ball Algorithm outperforms the Tree Algorithm on a grid; the Tree Algorithm performs better than the Ball Algorithm on a balanced tree; and on an Erdos-Renyi graph, the performances are similar, with the Ball Algorithm performing slightly better. We accomplish this by determining the probability of error for a range of infection times. We call an algorithm superior if it works in a wider range of times.

We note that to perform our simulations, it was necessary to use an approximate Steiner tree algorithm to perform the Tree Algorithm in a reasonable time frame. Naturally, since the exact problem is NP-hard, this would be required in any practical use of this algorithm at the moment. However, as a consequence, the empirical results may differ from the true theoretical result that would be obtained by employing an exact algorithm. Nevertheless, approximation algorithms typically have reasonable performance and we do not expect significant deviation from the correct results. The approximation algorithm we use is the Mehlhorn 2-approximation algorithm provided by the Goblin library [19]. This algorithm is an efficient algorithm which produces a Steiner tree with no more than twice the optimal number of edges.

Each of the points in these results represents the average of $10,000$ runs. The average infection size, which is used to normalize the expected infection size in a random sickness, was determined by averaging the results of $10,000$ infections. For each simulation, we use a reporting probability $q = 0.25$, and other parameters ($n$, $t$ and $m$) as specified in each section below. Finally, the graphs are plotted with error bars at 95% confidence.

## 6.1 Error Rate Versus Graph Size

Though our theoretical results have characterized the range for which each algorithm works, naturally we wish to see empirically the error probability for each algorithm and the rate at which the error decreases as graph size increases. Both Type I and Type II error probabilities were determined for each algorithm and graph topology. For this section, we have chosen time to keep the fraction of infected nodes at a consistent scaling. In particular, $t = 0.2\sqrt{n}$ for the grid, and $t = 0.5\log(0.5n)$ with $p = 2/n$ for the Erdös-Renyi graph.

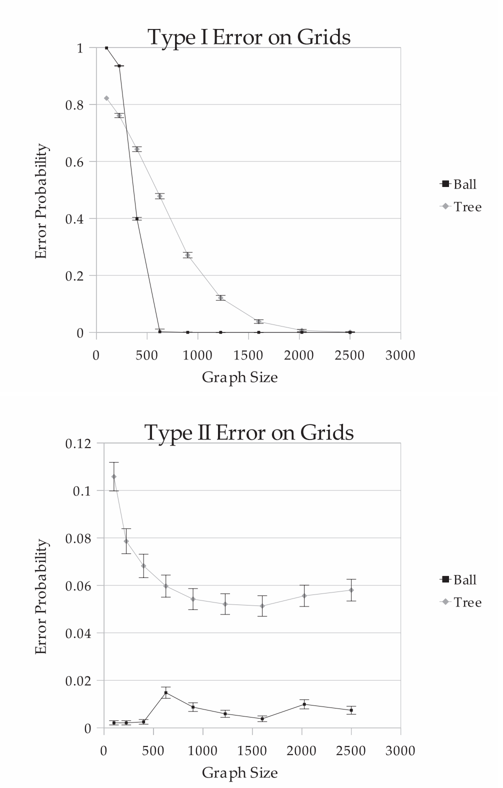

**Figure 3: Empirical Type I and Type II error probability vs graph size for grid graphs. The sample size is $10,000$ and infection size scales linearly with $n$.**

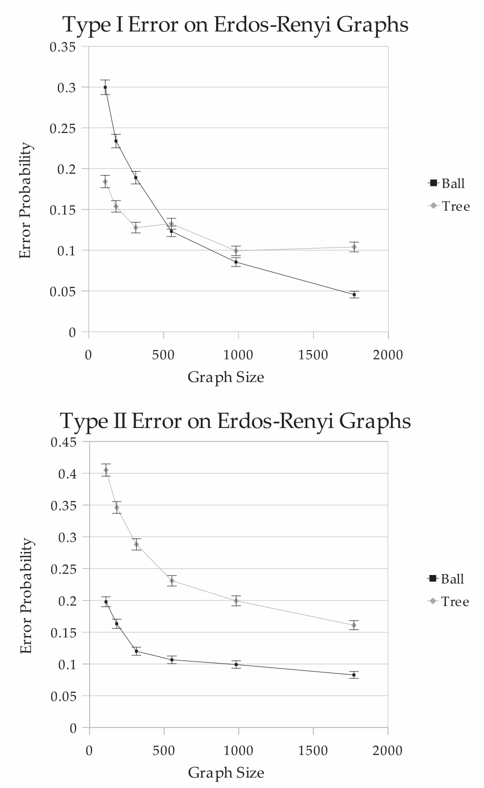

**Figure 4: Empirical Type I error probability vs graph size for graphs $G(n, 2/n)$. The sample size is $10,000$ and infection size scales orderwise as $\sqrt{n}$.**

The exact constants for these scalings were chosen empirically so that the problem was tractable, and the Type I and Type II errors were as balanced as possible. The thresholds $m$ were also chosen with the same scaling, according to our theoretical results. To be exact, for the grid, the Ball algorithm used threshold $m = 0.75\sqrt{n}$ and the Tree algorithm used threshold $m = 0.28n$. For the Erdös-Renyi graphs, the Ball algorithm used threshold $m = 0.69 \log(4.33n)$ and the Tree algorithm used threshold $m = 0.03\sqrt{n \log n} \log n$.

Figure 3 presents our results for grid graphs. The error probability of the Ball Algorithm on a grid is very low, while the tree algorithm performs relatively poorly. This is expected since the Ball Algorithm is closely aligned with the true shape of an infection on this graph. The Tree Algorithm has a much higher error probability which decays slowly with $n$, in particular Type II error.

Next, the results for Erdös-Renyi graphs are in Figure 4. Here we see again that the Ball Algorithm performs better than the Tree Algorithm, at least for larger $n$, and that the error probability also seems to be decreasing faster for the Ball Algorithm as well. Though a tree more closely matches the infection shape on an Erdös-Renyi graph, it is also easier for a random sickness to mimic a small tree, especially for small world graphs like Erdös-Renyi graphs. This causes the Ball Algorithm to be ultimately superior. The Tree Algorithm is superior for larger infection sizes on bottle necked graphs (such as trees) where the random sickness can be easily distinguished, as we see in Section 6.2.

## 6.2 Error Rate Versus Infection Time

Next, we examine empirically how the infection duration affects the probability of error for each of our algorithms. Since we are comparing the two algorithms by the range of infection sizes for which they work, we say an algorithm is superior if it maintains a lower probability of error for a wider time frame. We use thresholds that empirically minimize the overall probability of error. That is, the sickness was chosen to be either an infection or simply random with equal probability, and the threshold with minimum probability of error from the simulations was chosen.

These results are presented in Figure 5 for grids, trees, and Erdös-Renyi graphs. For each of the graph topologies, we used a graph size of $n = 1600$. The error probability is plotted against the average infection size from the simulation for a variety of infection times. This choice better conveys how infection size affects the error rate, which is the chief question of interest.

These charts allow us to compare the performance of the algorithms. It is clear that the error probability of the Ball Algorithm is less than that of the Tree Algorithm on both the grid and Erdös-Renyi graphs. On these graphs, the Ball Algorithm performs uniformly better at all timescales. However, the results on a tree are more complex. For low time scales where the infection is small, the Ball Algorithm has superior performance. However, for higher time scales, the Tree Algorithm has better performance. The larger timescales represent a larger infection, which would generally be considered a harder problem, especially since the

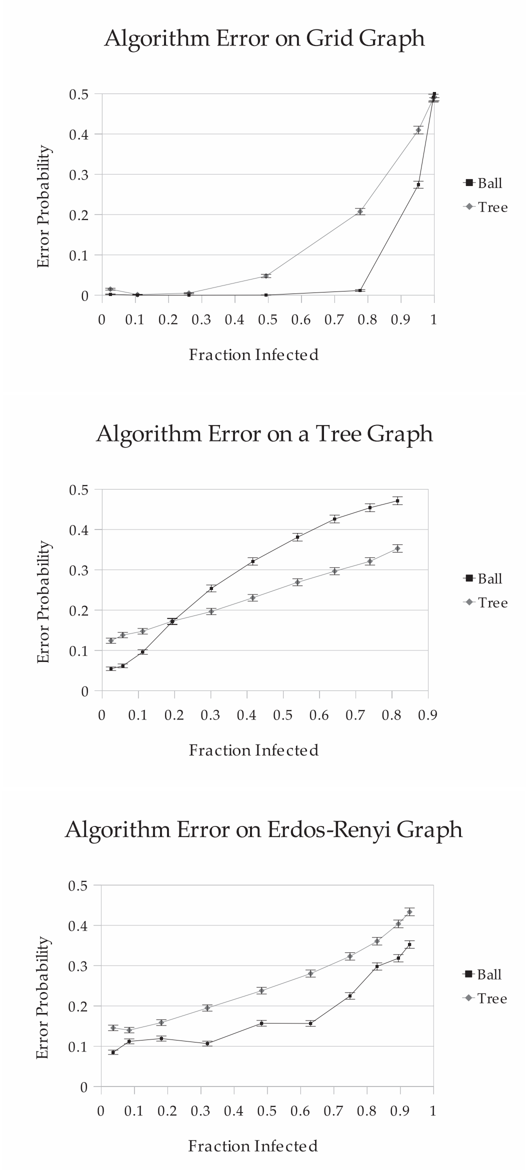

**Figure 5: This figure shows the overall error probability for each algorithm, for each of the three topologies we consider.**

infection is likely to have reached some of the leaves by this time. Then the Tree Algorithm is superior for this graph under this viewpoint. In addition, the Tree Algorithm maintains a fairly low error for a wider range of infection times. However, many practical applications of these algorithms would occur when the infection is still of limited size, in which case the Ball Algorithm would perform better. The best algorithm would depend on the circumstances.

It is particularly interesting to ask how these results extend to real-world graphs, as opposed to random (or highly regular) graphs that we have constructed. To this end, we used the call-graph from an Asian telecom network. In this graph, each node is a cell customer, and there is an edge between two users if they contacted each other over this network during a certain range of time. Since the original graph was too large for practical simulation times, we cut out a partial subset. We chose a random node and all nodes with a distance 9 and used the induced subgraph generated by these nodes. The resulting graph had size $n = 13189$. The probability of error for a range of times are present in Figure 6. We see that the results are similar to those for a Tree graph, where the Ball algorithm performs better on small infections, but it is out performed by the Tree algorithm at larger times. In particular, we see that the Ball algorithm performs particularly poorly for larger values of the infected fraction of nodes. This is to be expected, as the intuition for the Ball algorithm stems from the geometry of spatial grid-like networks. The call-graph here is very much tree-like (however, with very small diameter and high degree), and infections are unlikely to propagate to the same depth across various leaves. This will result in poor Ball "fits", especially as the infected fraction of nodes grow. This intuition is indeed borne out in the simulations.

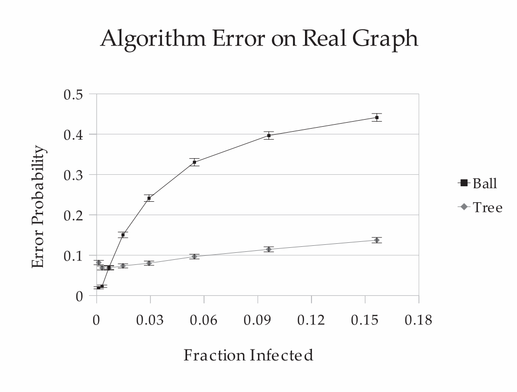

**Figure 6: This figure shows the overall error probability for each algorithm on a real world graph.**

The exact parameters for each point in the plots in Figure 5 and 6 are as follows.

*Grid graph:* (% infected, time ($t$), Ball threshold ($m$), Tree threshold ($m$)) – (2%, 2, 10, 40), (11%, 4, 21, 137), (26%, 6, 27, 267), (49%, 8, 31, 400), (78%, 10, 34, 525), (95%, 12, 35, 594), (99%, 14, 36, 631), (100%, 16, 37, 625).
*Tree graph*: (2%, 3, 4, 24), (6%, 4, 5, 48), (11%, 5, 5, 96), (19%, 6, 5, 160), (30%, 7, 5, 228), (42%, 8, 5, 296), (54%, 9, 5, 359), (64%, 10, 5, 425), (74%, 11, 5, 468), (82%, 12, 5, 508).
*Erdös-Renyi graph*: (4%, 3, 7, 35), (8%, 4, 8, 80), (18%, 5, 9, 156), (32%, 6, 10, 245), (48%, 7, 11, 337), (63%, 8, 11, 400), (75%, 9, 11, 459), (83%, 10, 12, 496), (89%, 11, 12, 513), (93%, 12, 12, 533).
*Real world graph*: (0.1%, 3, 7, 25), (0.3%, 4, 8, 87), (0.7%, 5, 8, 180), (1.4%, 6, 8, 350), (2.9%, 7, 8, 591), (5.4%, 8, 8, 959), (9.6%, 9, 8, 1396), (15.6%, 10, 8, 1896).

## 7. CONCLUSIONS

In this paper, we seek to answer the question: given a set of people reporting a sickness, how can you distinguish between a sickness spreading like an epidemic, or simply a illness that is occurring at random? To answer this question, we develop two natural algorithms which rate the likelihood a sickness is an infection by either the size of smallest ball or smallest tree containing the reporting sick nodes. We provide theoretical guarantees on the range of infection times and other parameters for which these algorithms succeed for several standard graph topologies. These have been supported by simulation data that provide insight into the true probability of error. As a high-level summary of our results, our work demonstrates both theoretically and empirically that the ball algorithm is superior on a grid, and the tree algorithm is superior on a tree graph.

There are several directions in which this work could be extended in the future. One natural direction is to ask the question, if you see the infection over a range of times, can you better estimate whether the sickness is a spreading epidemic or a simple random infection over the nodes? In particular, what is the best way to use the new information gained in this way? Another extension is considering that the sickness is spreading on one of two different networks (with potentially different epidemic models on each network), and we wish to determine which network the infection is traveling on.

## 8. ACKNOWLEDGMENTS

This work was partially supported by NSF Grants CNS-1017525, CNS-0721380, EFRI-0735905, EECS-1056028, DTRA grant HDTRA 1-08-0029 and Army Research Office Grant W911NF-11-1-0265. We also gratefully acknowledge Profs. Sujay Sanghavi and Sriram Vishwanath at The University of Texas at Austin for providing access to the (anonymized) call-graph data.

## 9. REFERENCES

[1] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders, “Social networks and context-aware spam,” in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, ser. CSCW ’08. ACM, 2008, pp. 403–412.

[2] NetworkX: http://networkx.lanl.gov.

[3] A. J. Ganesh, L. Massoulié, and D. F. Towsley, “The effect of network topology on the spread of epidemics,” in *INFOCOM*, 2005, pp. 1455–1466.

[4] F. Ball and P. Neal, “Poisson approximation for epidemics with two levels of mixing,” *The Annals of Probability*, vol. 32, no. 1B, pp. 1168–1200, 2004.

[5] A. Gopalan, S. Banerjee, A. K. Das, and S. Shakkottai, “Random mobility and the spread of infection,” in *INFOCOM 2011*, 2011, pp. 999–1007.

[6] N. Demiris and P. D. O’Neill, “Bayesian inference for epidemics with two levels of mixing,” *Scandinavian Journal of Statistics*, vol. 32, pp. 265–280, 2005.

[7] G. Streftaris and G. J. Gibson, “Statistical inference for stochatic epidemic models,” in *Proc. 17th International Workshop on Statistical Modeling*, 2002, pp. 609–616.

[8] N. Demiris and P. D. O’Neill, “Bayesian inference for stochastic multitype epidemics in structured populations via random graphs,” *Journal of the Royal Statistical Society Series B*, vol. 67, no. 5, pp. 731–745, 2005.

[9] D. Shah and T. Zaman, “Detecting sources of computer viruses in networks: Theory and experiment,” *SIGMETRICS Perform. Eval. Rev.*, vol. 86, pp. 203–214, 2010.

[10] ——, “Rumors in a network: Who’s the culprit?” *IEEE Transactions on Information Theory*, vol. 57, August 2011.

[11] H. Kesten, “On the speed of convergence in first-passage percolation,” *The Annals of Applied Probability*, vol. 3, no. 2, pp. 296–338, 1993.

[12] R. Lyons and R. Pemantle, “Random walk in a random environment and first-passage percolation on trees,” *The Annals of Probability*, vol. 20, no. 1, pp. 125–136, 1992.

[13] F. K. Hwang and D. S. Richards, “Steiner tree problems,” *Networks*, vol. 22, no. 1, pp. 55–89, 1992.

[14] C. Gröpl, S. Hougardy, T. NierHoff, and H. J. Proömel, *Approximation Algorithms for the Steiner Tree Problem in Graphs.* Kluwer Academic Publishers, 2000, pp. 235–279.

[15] I. Benjamini and Y. Peres, “Tree-indexed random walks on groups and first passage percolation,” *Probability Theory and Related Fields*, vol. 98, pp. 91–112, 1994.

[16] R. Durrett, *Random Graph Dynamics.* Cambridge University Press, 2007.

[17] F. Chung and L. Lu, “The diameter of sparse random graphs,” *Adv. in Appl. Math*, vol. 26, pp. 257–279, 2001.

[18] V. D. Blondel, J.-L. Guillaume, J. M. Hendrickx, and R. M. Jungers, “Distance distribution in random graphs and application to network exploration,” *Physical Review*, vol. 76, no. 066101, 2007.

[19] K. Mehlhorn, “A faster approximation algorithm for the steiner problem in graphs,” *Information Processing Letters*, vol. 27, pp. 125–128, 1988.