Fluid Modeling of Pollution Proliferation in P2P Networks

Rakesh Kumar* David D. Yao† Amitabha Bagchi‡ Keith W. Ross§ Dan Rubenstein¶

ABSTRACT

P2P systems are highly vulnerable to pollution attacks in which attackers inject multiple versions of corrupted content into the system, which is then further proliferated by unsuspecting users. However, to our knowledge, there are no closed-form solutions that describe this phenomenon, nor are there models that describe how the injection of multiple versions of corrupted content impacts a clients' ability to receive a valid copy. In this paper we develop a suite of fluid models that model pollution proliferation in P2P systems. These fluid models lead to systems of non-linear differential equations. We obtain closed-form solutions for the differential equations; for the remaining models, we efficiently solve the differential equations numerically. The models capture a variety of user behaviors, including propensity for popular versions, abandonment after repeated failure to obtain a good version, freeloading, and local version blacklisting. Our analysis reveals intelligent strategies for attackers as well as strategies for clients seeking to recover non-polluted content within large-scale P2P networks.

1. INTRODUCTION

By many measures, P2P file sharing continues to be one of the most important applications in the Internet today. There are currently more than 8 million users concurrently connected to either FastTrack/Kazaa, Gnutella, eDonkey2000 and eMule, with many additional users sharing files with BitTorrent. The content being shared includes MP3 songs, entire albums, television shows, entire movies, documents, images, software, and games. At the beginning of 2005, P2P constitutes about 60% of the Internet traffic in a tier-1 ISP [2]. In addition to file sharing, P2P is a promising architectural paradigm for distributed file systems [5, 13] and a variety of content distribution schemes [3, 8]

Nevertheless, many P2P systems are highly vulnerable to *pollution attacks*, in which attackers target specific content and inject corrupted versions of it into the system. Unable to distinguish polluted versions from unpolluted versions before actually downloading them, many unsuspecting users download polluted versions into their own file-sharing folders, from which other users may then later download the polluted versions [10, 4, 9]. In the Spring of 2005, pollution was highly prevalent in the FastTrack and eDonkey systems, with as many as 50 percent of copies of popular titles being polluted [9].

In this paper we develop a suite of fluid models for pollution proliferation in P2P systems. The models explicitly account for the presence of multiple versions for popular titles. Our fluid models allow us to investigate pollution proliferation for a variety of version-selection user behaviors. The fluid models lead to systems of non-linear differential equations. For two important version-selection behaviors - copy centric and version centric – we obtain closed-form solutions of the differential equations. We validate the fluid approximation with discrete-event simulation.

Later in the paper we integrate important real-world behaviors such as freeloading and user abandonment after repeated failures to download a non-polluted version. These behaviors can again be captured with fluid models. Where analytical results cannot be derived we efficiently solve the differential equations numerically. We also analyze more complex peer behaviors such as non-linear bias toward popular versions and use of anti-pollution strategies like local blacklisting and their implications on pollution attacks.

Our paper makes several novel contributions in the analysis of pollution proliferation in P2P systems, which, to our knowledge, has not been addressed in previous analytical work:

- We account for multiple *versions* of a file, as existing in today's P2P systems. When a client searches for a file, they are returned a list of versions, with the number of users offering that version. Our models enable us to consider the impact of pollution for different selection behaviors employed by the client, as well as how a polluter uses its resources to circulate different versions for the file it wishes to pollute.

- We obtain closed-form solutions that not only allow for faster analysis, but are more easily applied in future work that may require pollution models. Also,

*Dept. of ECE, Polytechnic University, Brooklyn, NY, USA, Email: rkumar04@utopia.poly.edu

†IEOR Dept., Columbia University, New York, NY, USA, Email: yao@columbia.edu

‡Dept. of Computer Science and Engineering, IIT Delhi, New Delhi, India, Email: bagchi@cse. iit.ac.in

§Dept. of Computer and Information Science, Polytechnic University, Brooklyn, NY, USA , Email: ross@poly.edu

¶Dept. of Electrical Engineering, Columbia University, New York, NY, USA, Email: danr@ee.columbia .edu

these solutions often permit us to explore the asymptotic regimes of these systems, as the number of clients continues to grow.

- We consider several practical variations of client behavior, including abandoning download after too many polluted copies are received, freeloading (not offering downloaded copies to other clients), version bias, blacklisting of known polluted versions, and non-negligible delay in download.

From our analysis, we are able to offer some preliminary important insights into pollution proliferation with multiple versions. We show that a polluter with limited resources is better off injecting a single polluted version into the network in such a way that this polluted version becomes very popular and spreads quickly. Clients are therefore better off selecting a version without considering its popularity. This is unless they can create a blacklisting mechanism where clients exchange information on bad versions. To counter such a blacklisting mechanism, a polluter must generate multiple polluted versions. Since this prevents any particular polluted version from becoming very popular, clients with a blacklisting mechanism should download taking into account the version popularity.

1.1 Related Work

There have been a number of measurement studies on pollution in P2P systems. In [10] the authors develop a crawler for the FastTrack network and report on the pervasiveness of pollution for popular content in the network. In [4] the authors examine intentional and unintentional pollution by conducting a measurement study of content availability in the eDonkey, FastTrack, Ares and Gnutella P2P networks. In [9] the authors report on pollution and index poisoning levels for Overnet and FastTrack.

In [6] the authors develop a set of difference equations modeling pollution proliferation. Only a small portion of the paper (Section III) is devoted to the type of file pollution attacks that is the focus of our paper, with the majority devoted to network oriented attacks which lack epidemiological properties of pollution attacks. That paper presents graphical results showing proliferation of pollution based on difference equations. Our work differs from [6] in a number of important ways. First we are able to derive closed form solutions for pollution proliferation for many important cases of peer behavior (unlike [6] which primarily relies on numerically solving the set of difference equations to gain insight). Second, [6] presents analysis assuming existence of a single good and single polluted version in the P2P network. However, [10, 4, 9] report existence of tens of thousands of polluted versions for a single title in the network. We present analysis and simulation results for multiple polluted and good versions. Our findings show that in certain situations success of a pollution attack depends strongly on the introduction of multiple polluted versions. Third, we present comprehensive fluid models which integrate important features like non-zero file download delay, higher bias towards popular versions, lack of altruism and peer abandonment.

There also exist interesting parallels between dynamics of worm propagation in the Internet and pollution proliferation in P2P networks. Both display epidemic behaviors wherein the rate of further infection depends on the current extent of infection in the network. However, pollution proliferation is not dependent on any underlying network topology, since users typically select versions from a nearly complete list of available versions in the P2P system. On the other hand, propagation of some worms (e.g, e-mail based) is influenced heavily by the social links of nodes involved [15]. Propagation of other worms (e.g., worms that use random scanning of IP addresses) display similar propagation behavior as of pollution and is independent of the underlying topology [7].

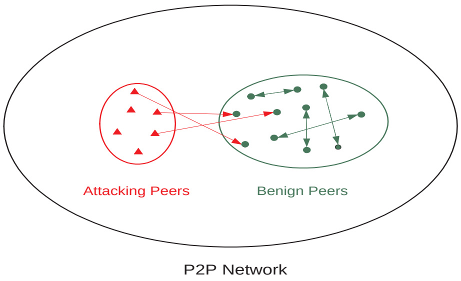


Figure 1. *The Attacker Nodes and the P2P Network.*

The paper proceeds as follows. Section 2 introduces our basic fluid model of the P2P system with pollution attacks. Sections 3 and 4 present in-depth analyses of the copy and version centric user behaviors respectively. Section 5 explores some practical variants of these models, including peers giving up on their attempts to download and peers who download but do not upload. Section 6 further extends the model to explore more complex user and system behaviors, such as increased bias toward popular versions, blacklisting of known polluted versions, and non-negligible download times. Section 7 concludes the paper.

2. BASIC MODEL

We first need to introduce some terminology. In this paper we investigate attacks against specific content in a P2P system. We shall refer to specific content – such as a specific song, movie, software, document – as a *title*. A given title can have many different versions. For music and video, these versions primarily result from the presence of a large number of rippers/encoders, each of which can produce a slightly different version of the same title. Furthermore, many file types include metadata embedded in the file (such as the ID3 tags in MP3 files), so that additional file versions are created when this metadata is modified. For a popular title, a P2P system may contain thousands of different versions. Each version has an identifier, which is typically a hash of the version.

Peers that actively introduce polluted copies into the P2P system are called *attacking peers*. The remaining peers are called *benign peers*. Figure 1 shows the interactions between the attacking peers and the benign peers. The interaction between the attacker peers and the benign peers is one-way signifying the fact that attacking peers offer polluted copies for download but never query for anything in the network. In contrast, the interactions between benign peers in the P2P network are two-way, implying peers both download from the network and serve other requests. Henceforth, a "peer" is a benign peer, unless otherwise indicated.

Throughout this paper we are concerned with how a single title (which can be a song, movie, TV show, game, book, software package, etc.) proliferates in the P2P network. We model and analyze the proliferation of both polluted and good versions of the given title. As we shall see, there is an intricate relationship between the proliferation of the good versions with the proliferation of the polluted versions.

In this section we describe our basic fluid model, where the "fluids" are the numbers of good and bad versions of content present in the P2P system, thus, these quantities are continuous functions of time. In subsequent sections we will examine both special cases and generalizations of this basic model. Let M be the number of benign peers that want to obtain a copy of the title. This title may have several "versions," some of which may be polluted. Each version may have several copies available in the network. When a peer queries for the title, the P2P application returns to the peer a list of all versions currently available. The peer has no advance knowledge of a version's quality; it determines the version's quality only after having downloaded and inspected the version. Initially we will make the following natural assumptions about peer behavior:

1. Once a peer obtains a good version, it stops searching for the title.
2. When inspecting a downloaded version, if the peer determines that the version is polluted, it deletes the version and immediately issues a new search query for the title. (In Section 5 we will relax this assumption, allowing peers to abandon trying to download a good version – with some probability – after determining that a downloaded version is polluted.)
3. When a peer has a good version, it makes it available indefinitely in the P2P network for uploading to other peers. (In Section 5 we remove this assumption, modeling the freeloading problem in P2P systems [1, 11]).
4. Peers are homogeneous, with all peers having the same behavior.

Thus, in our model at any time instant, the peers can be partitioned into three sets: (1) peers with a good copy; (2) peers with a polluted copy; (3) and peers with no copies. Note that each of the M peers possesses at most one copy (good or polluted) at any time.

After issuing a request for file download, typically a user physically leaves its peer device or engages in some other activity. We call the time from when the user issues a download request until the user returns to inspect the download result as the *inspection time*. The inspection time is a random quantity; let a peer's average inspection time be denoted by $1/\mu$. We refer to μ as a peer's *inspection rate*. According to our assumptions of peer behavior, if at inspection time the peer discovers the downloaded version to be polluted then it deletes version and issues a new search query for the title. The inspection rate is an important factor influencing the dynamics of pollution proliferation, since this is the rate at which a given peer deletes polluted versions and issues new search queries. In this section, we assume that the actual downloading time is negligible compared to a user's average inspection time. With high-speed residential access, the amount of time required to download an MP3 typically takes from tens of seconds to a few minutes. Thus, this assumption is reasonable for relatively small files, such as MP3s or short video clips. In Section 5 we generalize the model to allow for non-negligible download times, which will account for larger files.

A peer initially becomes aware of the title through the media (television, radio, etc.), a newsgroup, an e-mail, a website and so on. We make the natural assumption that a peer without a copy of the title issues its first query for the title with rate μ. (Although it is reasonable to assume that the inspection rate and the first-request rate are the same, the model remains tractable – albeit significantly more complicated – when using two distinct rates, say μ and λ.)

Let $\mathcal{V}(t)$ denote the set of versions (polluted and good) present in the network at time t. For a given version $v \in \mathcal{V}(t)$, denote the number of copies by $n_v(t)$. Because each peer has at most one copy of any version at any instant, $n_v(t)$ is also the number of peers with a copy of version v at time t.

After a user queries for a title, it receives a list of all versions of the title available in the network. [1] From this list of versions, the user chooses to download one version. In most P2P file sharing systems today, the user interface indicates how many copies of each version are available. The interface may also provide some estimate of the upload bandwidth of a version, where the bandwidth is aggregated over all copies of the version. When a user chooses a version for downloading, a user's choice may be biased towards versions with more copies. In general, the probability of selecting a particular version v can be modeled as a function of number of copies for each available version in the system:

$$q_v(t) = f_v(n_u(t), u \in \mathcal{V}(t)), \qquad v \in \mathcal{V}(t) \tag{1}$$

wherein $f_v(.), v \in \mathcal{V}(t)$ are arbitrary functions such that

$$\sum_{v \in \mathcal{V}(t)} q_v(t) = 1$$

To gain insight into this complex problem, we initially study the proliferation of pollution for two extreme cases of the selection distribution $q_v(t)$, $v \in \mathcal{V}(t)$.

1. *Copy Centric Model:* In this case, we suppose that users select for downloading a copy at random, uniformly across all copies available in the network. This is equivalent to supposing that a user chooses a version with a probability in proportion to the number of copies of that version available in the system, that is,

$$q_v(t) = \frac{n_v(t)}{\sum_{u \in \mathcal{V}(t)} n_u(t)}, \qquad v \in \mathcal{V}(t) \tag{2}$$

2. *Version Centric Model:* In this case, we take the versions explicitly into account, and suppose users select a version for downloading independently of the number of copies of the versions available, that is,

$$q_v(t) = \frac{1}{|\mathcal{V}(t)|}, \qquad v \in \mathcal{V}(t) \tag{3}$$

For both Copy Centric and Version Centric Models we present the analysis for when (i) the P2P network is under a pollution attack and (ii) when it is recovering from the attack.

[1] For P2P systems with centralized directories – such as Napster – or systems with a DHT structure, this assumption is quite reasonable. For some unstructured P2P systems, the user may receive only a partial list. For the purpose of constructing an insightful and tractable model, we assume that peer becomes aware of all versions currently in the network.

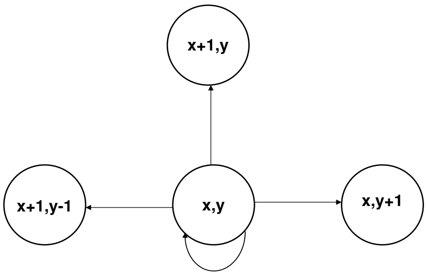


Figure 2. *State transitions for Markov process for the Copy Centric Model.*

For most reasonable choices of the selection probability function $q_v(t)$, including for the Copy and Version Centric Models, eventually every peer will succeed in getting a good copy of the title. This is because, in the current model, peers with good copies make their copies available indefinitely to other peers and because peers repeatedly download until they receive a good copy. We relax these assumptions in Section 5 and study pollution proliferation in a network containing peers who are non-persistent and freeloading.

3. COPY CENTRIC MODEL

3.1 Under Pollution Attack

When the system is under attack, the attacker uses its own peers (or peers that it controls) to introduce polluted versions into the system. With the use of these attacking peers, let N denote the number of polluted versions the attacker presents to the network (which is assumed to be constant over time). Each of the M benign peers wants to obtain a copy of a good version of the title. A small fraction of the M peers obtain good versions of the title from outside the P2P system, for example, from ripping CDs, from Web-based software distributions, and so on.

One can deduce from equation (2) that the probability that a peer selects a polluted version for downloading is the ratio of the total number of polluted copies to the total number of polluted and good copies in the system combined. Thus, for the Copy Centric Model, we can ignore version types since they do not influence the dynamics of the pollution proliferation. We can instead simply focus on the total number of polluted copies and the total number of good copies for the Copy Centric Model.

Before presenting our fluid flow analysis of the system, we briefly describe the discrete-state Markov process approach for analyzing the pollution proliferation. At a given instant of time, let x and y denote of the number of benign peers with good and and polluted copies, respectively. Assuming a peer's inspection time is exponentially distributed with rate μ, the tuple (x, y) is a Markov process, with jumps occurring at inspection instants. Figure 2 shows the state transitions for the Markov process. As shown in this figure, from the state (x, y) the system can go to the state

1. $(x+1, y)$: a peer with no copy downloads a good copy at its first request.

2. $(x, y+1)$: a peer with no copy downloads a polluted copy at its first request.

3. $(x+1, y-1)$: a peer with polluted copy downloads a good copy.

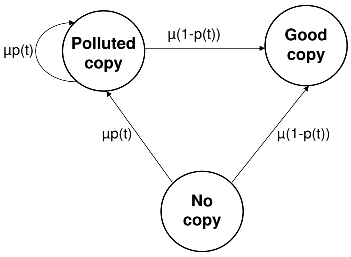


Figure 3. *State transition diagram.* $p(t)$ denotes probability of downloading a polluted copy at time t

4. (x, y): a peer with polluted copy downloads a polluted copy at its next request.

The rate of inspections is $(M - x)\mu$ when the process is in state (x, y). From equation (2) for any given peer request, the probability that a peer selects a polluted copy for download is

$$p = \frac{y + N}{x + y + N}.$$

It is a straightforward exercise to obtain the transition rates for the Markov process.

Ideally, we would like to solve for transient measures of the Markov process, such as the probability distribution for the time to reach the absorbing state $(M, 0)$ from any initial state. Unfortunately, this problem appears extremely unwieldy, since the cardinality of the state space is roughly M^2 and since M is very large, often in the tens or hundreds of thousands.

Given the intractability of the Markov model, we consider a fluid flow approximation of the system. In particular, let $x(t)$ and $y(t)$ denote the total number of peers with good copies and polluted copies, respectively, at time t. In the fluid model, we view $x(t)$ and $y(t)$ as continuous, deterministic quantities. As with the Markov model, at any given time t, the probability that a peer selects a polluted copy for download is

$$p(t) := \frac{y(t) + N}{x(t) + y(t) + N}.$$

The number of good copies, $x(t)$ increases when a peer with no copy downloads a good copy, which happens with rate $[M - x(t) - y(t)]\mu(1 - p(t))$, or when a peer with a polluted copy downloads a good copy, which happens with rate $y(t)\mu(1 - p(t))$. This leads to the fluid equation

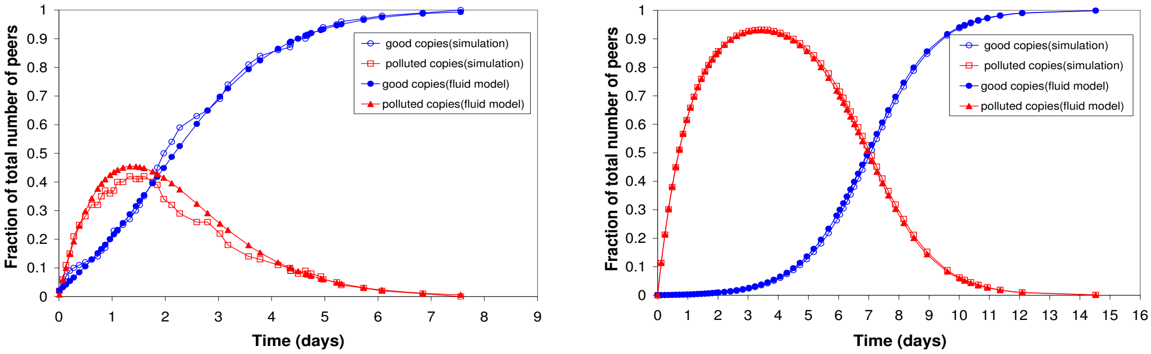
$$\dot{x}(t) = [M - x(t) - y(t)]\mu(1 - p(t)) + y(t)\mu(1 - p(t))$$

Similarly, the number of polluted copies, $y(t)$, increases when a peer with no copy downloads a polluted copy, which happens with rate $[M - x(t) - y(t)]\mu p(t)$; however, $y(t)$ decreases when a node with a polluted copy downloads a good copy, which occurs with rate $y(t)\mu(1 - p(t))$. This leads to the fluid equation

$$\dot{y}(t) = [M - x(t) - y(t)]\mu p(t) - y(t)\mu(1 - p(t))$$

Figure 3 shows the state transition diagram of this fluid system. After some simple algebra, the equations can be simplified to

$$\begin{aligned} \dot{x}(t) &= [M - x(t)]\mu(1 - p(t)) && (4) \\ \dot{y}(t) &= [M - x(t)]\mu p(t) - \mu y(t) && (5) \end{aligned}$$



(a) Small system ($M = 100$)

(b) Large system ($M = 100,000$)

Figure 4. *Model Validation: Validation of the fluid model*

Although this system of differential equations is nonlinear, we can nevertheless obtain a closed-form solution.

Theorem 1: The system of differential equations (4-5) has the following unique solution:

$$x(t) = \frac{c_2 M \left(e^{\mu t} - \frac{c_1}{M+N}\right)^{\frac{M}{M+N}}}{1 + c_2\left(e^{\mu t} - \frac{c_1}{M+N}\right)^{\frac{M}{M+N}}} \tag{6}$$

and

$$y(t) = M - c_1 e^{-\mu t} - x(t), \tag{7}$$

where

$$c_1 = M - x(0) - y(0) \tag{8}$$

and

$$c_2 = \frac{x(0)}{M - x(0)}\left(\frac{N + x(0) + y(0)}{M + N}\right)^{-\frac{M}{M+N}} \tag{9}$$

Proof: Let $y(t) + x(t) = z(t)$. Summing equation (4) and (5) gives

$$\dot{z}(t) = [M - z(t)]\mu$$

Solving this simple differential equation gives

$$z(t) = x(t) + y(t) = M - c_1 e^{-\mu t}$$

Substituting $p(t) = \frac{y(t)+N}{x(t)+y(t)+N}$ in (4) we have

$$\dot{x}(t) = [M - x(t)]\mu \frac{x(t)}{x(t) + y(t) + N}$$

Substituting result from (8) and integrating both sides we obtain

$$\int \frac{dx(t)}{x(t)(M - x(t))} = \mu \int \frac{dt}{M + N - c_1 e^{-\mu t}} + const.$$

$$\frac{1}{M} \ln\left(\frac{x(t)}{M - x(t)}\right) = \frac{1}{M+N} \ln\left(e^{\mu t} - \frac{c_1}{M+N}\right) + const.$$

Further simplifying the above expression we obtain

$$\frac{x(t)}{M - x(t)} = c_2 \left(e^{\mu t} - \frac{c_1}{M+N}\right)^{\frac{M}{M+N}}$$

for some constant c_2. Thus $x(t)$ can be written as:

$$x(t) = \frac{c_2 M \left(e^{\mu t} - \frac{c_1}{M+N}\right)^{\frac{M}{M+N}}}{1 + c_2\left(e^{\mu t} - \frac{c_1}{M+N}\right)^{\frac{M}{M+N}}}$$

The constant c_2 can be calculated by substituting values of $x(0)$ in (6) giving (9) □

Model Validation

We validated the fluid model with a discrete event simulation of the P2P network. We examined two scenarios:

- Small system: $M = 100$, $N = 10$, $x(0) = 2$, $y(0) = 0$, and $\mu = 1$ query/day.
- Large system: $M = 100,000$, $N = 10,000$, $x(0) = 20$, $y(0) = 0$, and $\mu = 1$ query/day.

For the two scenarios, Figure 4 compares the fluid solution obtained from Theorem 1 with a realization of a discrete-event simulation. For the small system, we see that the fluid approximation captures the general trend of the discrete event system. In both the approximation and the simulation, the number of polluted copies rises to a peak and then drops off exponentially to zero. (We will discuss in more detail the qualitative behavior of the system in Section 6.) For the large system, not only does the fluid approximation capture the general trend of the simulation, but it actually tracks the simulation very closely, providing an excellent approximation. Since P2P systems typically have a large number of peers interested in a popular title, large system models are more reflective of real-world scenarios.

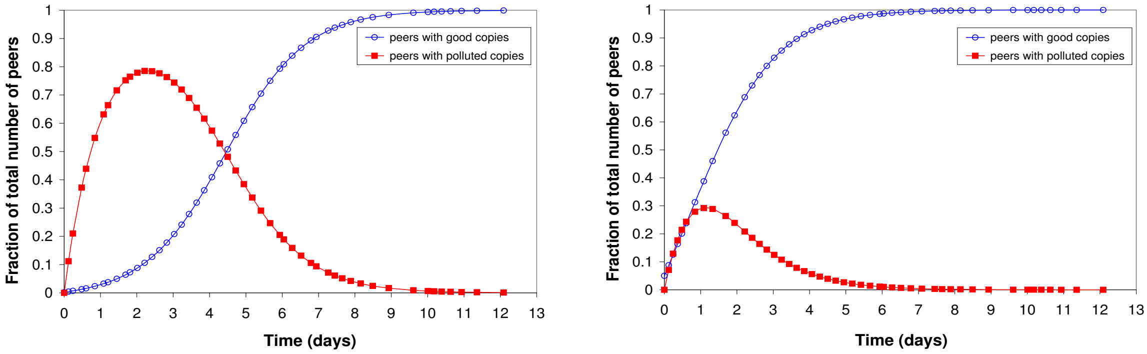
Impact of Initial Conditions

The initial ratio of polluted copies to good copies in the system,

$$K := \frac{y(0) + N}{x(0)},$$

is critical in determining how the pollution attack plays out in the system. With $M = 100,000$, we examine two cases for K. In the first case the initial number of peers with good copies is 200 and in the second case it is 5,000. In both cases, the number of polluted copies available, $y(0) + N$, is set to 10,000. Thus we examine K values of $K = 50$ and $K = 2$. These numbers are arbitrary and serve the purpose of illustrating the effect of initial ratio of good copies to polluted copies on pollution proliferation. In Figure 5 we plot pollution proliferation for these two cases. We see from this figure that for an attacker its a more effective strategy to launch the pollution attack in advance of the release date of the title. To analytically explain these results its helpful to study the proliferation of pollution in a very large system. To this end, consider

$$x_\infty(t) = \lim_{M\to\infty} \frac{x(t)}{M}, \qquad y_\infty(t) = \lim_{M\to\infty} \frac{y(t)}{M}$$



(a) $K = 50$.

(b) $K = 2$

Figure 5. *Effect of initial ratio of bad to good copies*

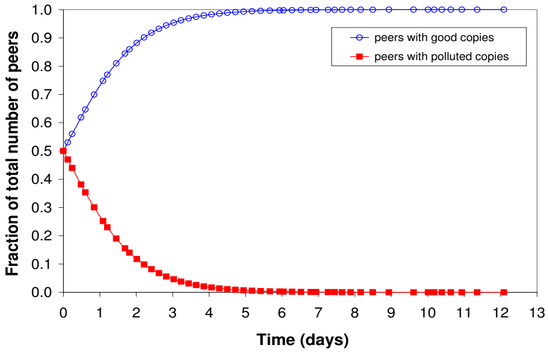


Figure 6. *Copy Centric Model: Recovery from a pollution attack*

From Theorem 1, it is straightforward to show

$$x_\infty(t) = \frac{1 - e^{-\mu t}}{1 + Ke^{-\mu t}}; \quad y_\infty(t) = \frac{Ke^{-\mu t}}{1 + Ke^{-\mu t}} \tag{10}$$

Observe from (10) that the initial number of polluted copies critically influences the pollution proliferation in a P2P network even when the number of peers interested in the title approaches infinity.

3.2 Recovering from a Pollution attack

The attacker at some stage may decide to stop offering polluted copies because it is more attractive to pollute some other title. In that case the only source of polluted copies in the network are the benign peers. In this case the relationships for $x(t), y(t)$ can be obtained by simply substituting $N = 0$ into Theorem 1:

$$x(t) = \frac{M - c_1 e^{-\mu t}}{1 + Ke^{-\mu t}} \tag{11}$$

$$y(t) = K\frac{M - c_1 e^{-\mu t}}{1 + Ke^{-\mu t}} e^{-\mu t} \tag{12}$$

with K now simply being $y(0)/x(0)$

In Figure 6 we show recovery of pollution corresponding to the system in Figure 5(a) with $y(0) = x(0) = 50,000$. One can observe that the time interval (t_0, T), where $x(t_0) = y(t_0) = 50,000$ and $x(T) = 100,000$, $y(T) = 0$ in Figure 5(a) is distinctly more than the time interval $(0, T)$, where $x(0) = y(0) = 50,000$ and $x(T) = 100,000$, $y(T) = 0$ in Figure 6. Thus, starting from the same initial conditions it takes less time for every peer to get a good copy in the pollution recovery case as compared to the pollution attack case. This is expected because $K = (50,000 + 10,000)/10,000 = 1.2$ at time instant t_0 in the former case of pollution attack and it is $K = 50,000/50,000 = 1$ at time instant 0 in the latter case of pollution recovery.

Now let t_ϵ be the time $x(t)$ becomes within ϵ of M, i.e., $x(t_\epsilon) = M - \epsilon$. We have,

$$M - \epsilon = \frac{M - c_1 e^{-\mu t_\epsilon}}{1 + Ke^{-\mu t_\epsilon}}$$

Thus,

$$t_\epsilon = \frac{1}{\mu} \ln \left(\frac{K(M - \epsilon) + c_1}{\epsilon} \right)$$

Hence, the time until the almost all of the peers have a good copy is

$$t_\epsilon = O(\frac{1}{\mu} \ln \frac{M}{\epsilon}) \tag{13}$$

We shall compare this to the corresponding result in the Version Centric model to gain insight on the fundamental differences in pollution proliferation for the two models.

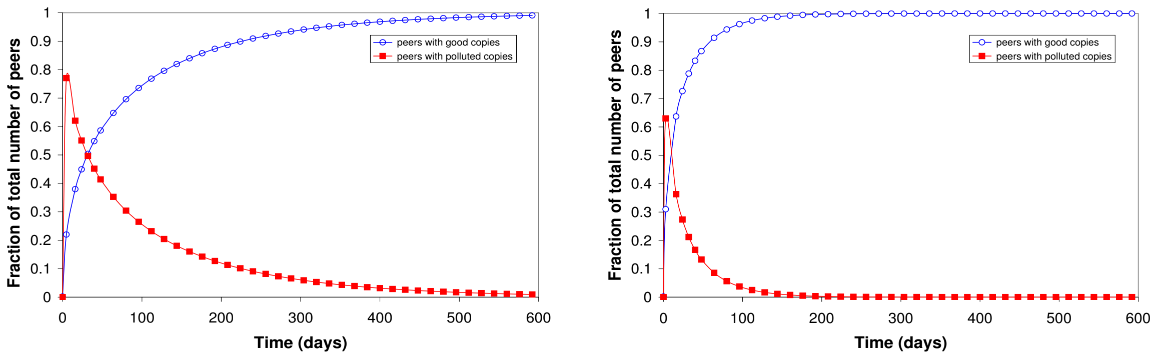
4. VERSION CENTRIC MODEL

Recall that in the Version Centric Model, we take different versions of a title explicitly into account, and assume that users select a version to download independently of the number of copies of the versions available. Furthermore, a user selects a version uniformly from the set of all available versions. This natural model turns out to be tractable analytically, as was the Copy Centric Model.

To fully describe this model, we need to specify how good and polluted versions are introduced into the system. We suppose that there are a constant number, g, of good versions of the title in the system. Let $w(t)$ denote the number of polluted versions available in the network (including both attacking peers and benign peers) at time t. Thus when a peer issues a download request at time t, the probability of downloading a copy of a good version is $g/(g + w(t))$; and the probability of downloading a copy of a polluted version is $w(t)/(g + w(t))$.

4.1 Under Pollution Attack

Given the user's disregard for a version's popularity, the optimal pollution attack strategy would be to offer an infinite number of polluted versions so as to make the probability of



(a) g = 25

(b) g=50

Figure 7. *Proliferation of pollution for Version Centric model: Pollution attack*

selection of a good version zero. However, let us assume that for lack of storage, bandwidth resources the attacker can only offer $w(0)$ polluted versions at any given time. Given that, the optimal attack strategy is to maximize the number of polluted versions in the system. To do this, an attacker would wait for a polluted version to be downloaded for the the first time then immediately replace it with a newly constructed (never downloaded) polluted version. This strategy maximizes the probability $w(t)/(g+w(t))$ that a benign peer selects a polluted version. Given this attacker strategy, the probability a *new* polluted version is introduced into the set of M benign peers upon a download request is $w(0)/(g+w(t))$. This fraction denotes the probability that a peer requests one of the latest polluted versions on offer from the attacker.

Note the number of good versions does not change through downloading since they are fixed to be g. What will change is the number of good *copies*, which we again denote as $x(t)$. In addition, again let $y(t)$ denote the total number of polluted copies over all the polluted versions. Also note that $y(t)$ is different from $w(t)$, which is the total number of polluted *versions* available in the network.

Once again, assuming exponential inspection times, the Version Centric Model can again be modeled as a Markov process with state $(x(t), y(t), w(t))$. However, its analysis becomes unwieldy for systems with a large number of peers, M, which is typically the case in practice.

For the fluid model, we view $x(t)$, $y(t)$ and $w(t)$ as continuous rather than discrete variables. Making arguments similar to those made for the Copy Centric Model, we obtain

$$\begin{aligned}
\dot{x}(t) &= [M - x(t) - y(t)]\mu\left[\frac{g}{g+w(t)}\right] + y(t)\mu\left[\frac{g}{g+w(t)}\right] \\
\dot{y}(t) &= [M - x(t) - y(t)]\mu\left[\frac{w(t)}{g+w(t)}\right] - y(t)\mu\left[\frac{g}{g+w(t)}\right] \\
\dot{w}(t) &= [M - x(t) - y(t)]\mu\left[\frac{w(0)}{g+w(t)}\right] + y(t)\mu\left[\frac{w(0)}{g+w(t)}\right]
\end{aligned}$$

After some simple algebra, we can rewrite these equations as:

$$\dot{x}(t) = [M - x(t)]\mu\left[\frac{g}{g+w(t)}\right] \tag{14}$$

$$\dot{y}(t) = [M - x(t)]\mu\left[\frac{w(t)}{g+w(t)}\right] - y(t)\mu \tag{15}$$

$$\dot{w}(t) = [M - x(t)]\mu\left[\frac{w(0)}{g+w(t)}\right] \tag{16}$$

Theorem 2: The number of good copies in the system, $x(t)$, satisfies the following implicit equation for each t:

$$x(t) = M - [M - x(0)]\exp\left[\frac{-x(t) + x(0) - \mu g^2 t/w(0)}{M - x(0) + g + g^2/w(0)}\right] \tag{17}$$

The number of polluted copies in the system, $y(t)$, satisfies the following relationship with $x(t)$:

$$\begin{aligned}
y(t) &= M - c_1 e^{-\mu t} - x(t) \\
c_1 &= M - x(0) - y(0)
\end{aligned}$$

Remark: Note for any given value of t, the implicit equations can be easily and rapidly solved numerically.
Proof: Adding (14) and (15) we get,

$$\dot{x}(t) + \dot{y}(t) = [M - x(t) - y(t)]\mu$$

Thus

$$x(t) + y(t) = M - c_1 e^{-\mu t}, \quad \text{with} \tag{18}$$

where $c_1 = M - x(0) - y(0)$

From (14) and (16), we have $\dot{w}(t)/w(0) = \dot{x}(t)/g$. Hence,

$$w(t) - w(0) = (x(t) - x(0))w(0)/g$$

or equivalently

$$x(t) = \frac{g}{w(0)}w(t) + d_0.$$

where $d_0 := x(0) - g$. Substituting this into (14), we have

$$\frac{w(0)(x(t) - d_0)/g + g}{M - x(t)}dx(t) = \mu g dt,$$

or

$$\left[-\frac{w(0)(M - d_0)/g + g}{M - x(t)} + \frac{w(0)}{g}\right]dx(t) = -\mu g dt;$$

and hence,

$$(w(0)(M - d_0)/g + g)\ln\frac{M - x(t)}{M - x(0)} + \frac{w(0)}{g}(x(t) - x(0)) = -\mu g t,$$

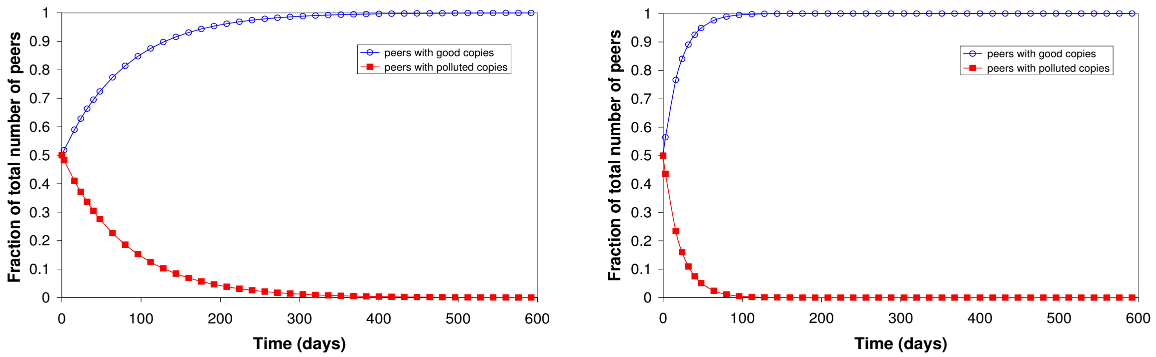
or

$$\frac{M - x(t)}{M - x(0)} = \exp\left[\frac{-w(0)(x(t) - x(0))/g - \mu g t}{w(0)(M - d_0)/g + g}\right].$$

Therefore, we have

$$x(t) = M - [M - x(0)]\exp\left[\frac{-x(t) + x(0) - \mu g^2 t/w(0)}{M - x(0) + g + g^2/w(0)}\right].$$

By substituting the result from equation (18) into (17), we also obtain the expression for $y(t)$. □



(a) g = 25

(b) g=50

Figure 8. *Proliferation of pollution for Version Centric model: Recovery*

As in the last section let t_ϵ be the time $x(t)$ has reached $M - \epsilon$. We have,

$$[M - x(0)] \exp \left[\frac{-M + \epsilon + x(0) - \mu g^2 t_\epsilon / w(0)}{M - d_0 + g^2/w(0)} \right] = \epsilon.$$

That is,

$$\frac{M - \epsilon - x(0) + \mu g^2 t_\epsilon / w(0)}{M - d_0 + g^2/w(0)} = \ln \frac{M - x(0)}{\epsilon}.$$

When M is large, the above simplifies to

$$\frac{\mu g^2 t_\epsilon}{w(0)} \sim M \ln \frac{M}{\epsilon}.$$

Hence,

$$t_\epsilon = O(\frac{M}{\mu} \ln \frac{M}{\epsilon}) \tag{19}$$

Comparing the above with (13), we note the additional factor M. From equations (14) and (16) we note that here the rate of growth of polluted versions is proportional to the rate of growth of good copies, whereas the number of good version stays at the constant g. Consequently, as time goes by, it becomes increasingly more difficult to get a good copy. However, eventually, i.e., when time is large in the sense of (19), almost everyone will have obtained a good copy.

Because of the different asymptotics of t_ϵ, the behavior of $x_\infty(t) := \lim_{M\to\infty} x(t)/M$ is also different from the copy-centric model. Fix t and divide both sides of (17) by M. Then, when $M \to \infty$, the equation in (17) is reduced to

$$x_\infty(t) = 1 - e^{-x_\infty(t)},$$

with the solution being $x_\infty(t) = 0$. This is in sharp contrast with the result in (10).

However, suppose we let $t \to \infty$ along with $M \to \infty$, while maintaining the ratio t/M as a constant, specifically,

$$\frac{\mu g^2 t}{w(0)M} \to c, \tag{20}$$

with c being some positive constant. Denote the ratio $\lim_{t,M\to\infty} x(t)/M := r$ in this case. Then, as $t, M \to \infty$, (17) becomes

$$r = 1 - e^{-(c+r)}.$$

The above equation must have a unique solution $r^*(c) \in (0, 1)$. [This is because the function $f(r) := 1 - r - e^{-(c+r)}$ is decreasing as $f'(r) = -1 + e^{-(c+r)} < 0$, and $f(0) = 1 - e^{-c} > 0$, $f(1) = -e^{-c+1} < 0$.] This ratio, $r^*(c)$, is the asymptotic analogue to $x_\infty(t)$ in (10) of the Copy Centric Model. As an example, $r^*(1) = 0.8414$. Furthermore, if $t = O(M)$ then $r^*(1) < 1$ but if $t = O(M \ln(\frac{M}{\epsilon}))$ then from (19) $r^* = 1$.

In Figure 7 we plot pollution proliferation by numerically solving the implicit equations in Theorem 2. The initial conditions for the simulation setup are: $x(0) = 200$, $w(0) = 1$, $y(0) = 0$, $M = 100{,}000$ peers. We vary the number of good versions in the system with $g = 25$ and $g = 50$ and show the corresponding proliferation curves in Figure 7(a) and 7(b), respectively. It can be observed that a higher number of good versions in the system critically influences the pollution proliferation curves and results in a shorter time by which every peer gets a copy of a good version. Also it is instructive to reflect on the time required for every peer to get a good copy which is an order of magnitude more than the same time required in the Copy Centric Model. Thus, a version selection behavior wherein peers completely ignore relative popularity of versions will work to the attacker's advantage.

4.2 Recovering from pollution attack

Denote by b the number of polluted versions in the system when the attacker stops actively polluting the network. It is easy to see that from then on the probability of downloading a polluted version becomes $p(t) = b/(b+g)$. Thus, the state equations are:

$$\dot{x}(t) = [M - x(t)]\mu \left[\frac{g}{g+b} \right] \tag{21}$$

$$\dot{y}(t) = [M - x(t)]\mu \left[\frac{b}{g+b} \right] - \mu y(t) \tag{22}$$

Further from (21) we have,

$$\int \frac{dx(t)}{M - x(t)} = \int \frac{\mu g}{g+b} dt$$

$$\ln \frac{M - x(t)}{M - x(0)} = -\frac{\mu g}{g+b} t$$

Thus, for this case we have the following closed-form solution:

$$x(t) = M - [M - x(0)] \exp \left[-\frac{\mu g}{g+b} t \right] \tag{23}$$

Notice that (18) will still hold with the above system of equations. Thus using results from (18) and (23), we have

$$\begin{aligned} y(t) &= M - [M - x(0) - y(0)] \exp[-\mu t] - \\ & \quad [M - x(0)] \exp \left[-\frac{\mu g}{g+b} t \right] \end{aligned} \tag{24}$$

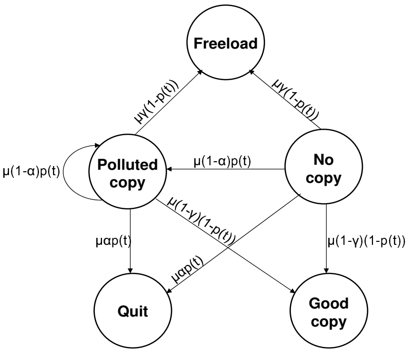


Figure 9. *State transition diagram for abandonment and freeloading*

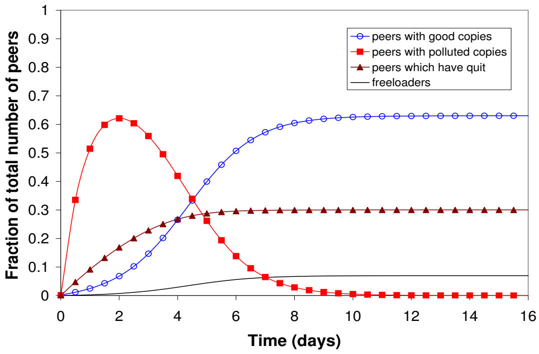


Figure 10. *Proliferation with abandonment and freeloading*

We now compare the pollution recovery curves for the pollution attack instances shown in Figure 7. We assume the attacker stops the attack at the time instant t_0 when $x(t_0) = y(t_0)$. For the first case of $g = 25$ shown in Figure 7(a), $w(t_0) = b = 1996$, $x(t_0) = y(t_0) = 50,000$. The result of plugging this into (23) and (24 is plotted in Figure 8(a). The corresponding values for the case with $g = 50$ are $w(t_0) = b = 1002$, $x(t_0) = y(t_0) = 50,000$. We plot pollution recovery curves in Figure 8. Comparing these figures to their corresponding proliferation figures in the attack phase, we see that it takes considerably less time for every peer to get a good copy in the recovery phase than in the attack phase. This is to be expected because the probability of selection of a good version is constant in the recovery phase while it is a non-decreasing in the attack phase. Further, it is easy to see that the recovery time will be longer if the attack lasts longer.

5. MODELING PEER ABANDONMENT AND FREELOADING

So far we have assumed that peers are fully persistent in their attempts to download a good copy. In practice, however, if a peer repeatedly fails to download a good copy, it will become frustrated and abandon downloading. Indeed this is one of the attacker's goals: to frustrate the peers in their attempts to get a good copy, so that in future they are discouraged from trying to download any other content. We model peer abandonment as follows: Whenever a peer discovers that its last request resulted in downloading a copy of a polluted version, then with probability α it quits and leaves the network. Thus $\alpha = 1$ corresponds to the case of peers giving up after receiving their first polluted copy, whereas $\alpha = 0$ corresponds to fully persistent peers. We classify the set of peers who have given up before downloading a good copy as being in the state "quit".

Another assumption we have made is that peers always share their downloaded copies. This may not always be a realistic assumption as lack of altruism in P2P networks has been widely reported [1]. To take into account freeloaders, we can focus on the peers not sharing the good copies because polluted copies are shared freely before they are deleted. This can be explained as follows: typically a file is downloaded into the user's default shared directory. A downloaded polluted copy will be freely available until it is discovered, at which time it is deleted. Thus it suffices to model freeloading in terms of peers that decide to not share their good copies. To capture freeloading in our model, we suppose that a peer that discovers that it has obtained a good copy decides to not share it with probability γ. For all practical purposes, the freeloading peer has "left" the network since it is neither sharing the downloaded good copy nor is interested in downloading another copy. Thus $\gamma = 1$ will imply the peers do not cooperate at all.

Figure 9 shows the expanded state transition diagram, which models both peer abandonment (state "quit") and freeloading. Let $q(t)$ and $f(t)$ denote the number of peers which are in state "quit" and "freeload" at time instant t, respectively. Let $x(t)$,$y(t)$ and $p(t)$ have the same meanings as in section 3. Then the fluid equations become:

$$\begin{aligned}
\dot{y}(t) &= [M - x(t) - y(t) - q(t) - f(t)]\mu(1-\alpha)p(t) - \\
&\quad y(t)[\mu\alpha p(t) + \mu\gamma(1-p(t)) + \mu(1-\gamma)(1-p(t))] \\
\dot{x}(t) &= [M - x(t) - y(t) - q(t) - f(t)]\mu(1-\gamma)(1-p(t)) + \\
&\quad y(t)[\mu(1-\gamma)(1-p(t))] \\
\dot{q}(t) &= [M - x(t) - y(t) - q(t) - f(t)]\mu\alpha p(t) + \\
&\quad y(t)\mu\alpha p(t) \\
\dot{f}(t) &= [M - x(t) - y(t) - q(t) - f(t)]\mu\gamma(1-p(t)) + \\
&\quad y(t)\mu\gamma(1-p(t))
\end{aligned}$$

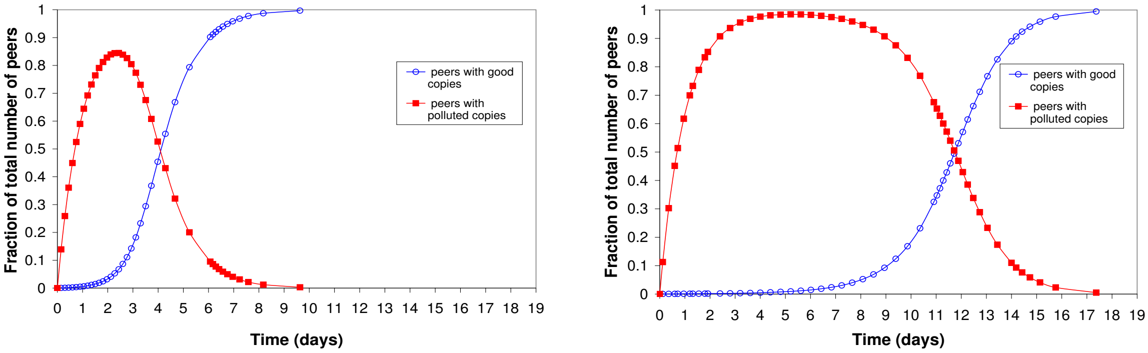
Simplifying the above system of equations, we have

$$\begin{aligned}
\dot{y}(t) &= [M - x(t) - q(t) - f(t)]\mu(1-\alpha)p(t) - \mu y(t) && (25) \\
\dot{x}(t) &= [M - x(t) - q(t) - f(t)]\mu(1-\gamma)(1-p(t)) && (26) \\
\dot{q}(t) &= [M - x(t) - q(t) - f(t)]\mu\alpha p(t) && (27) \\
\dot{f}(t) &= [M - x(t) - q(t) - f(t)]\mu\gamma(1-p(t)) \\
&= \frac{\gamma}{1-\gamma}\dot{x}(t) && (28)
\end{aligned}$$

It appears difficult to obtain a closed-form solutions for these non-linear differential equations. However, they can be efficiently solved numerically. For example, let $\alpha = 0.1$ (average number of 10 queries per peer) and $\gamma = 0.1$ (only 1 in 10 peer freeloads). With these parameters, we numerically solve the differential equation to obtain Figure 10. In this figure, the initial number of peers with good copies, $x(0)$, is 200 and the initial number of polluting peers, N, is 10,000. On comparing this to the corresponding Figure 5(a) (with $\alpha = \gamma = 0$) it is easy to see that even conservative values of $(\alpha = \gamma = 0.1)$ influence the fraction of peers with good copies drastically which converges to a value strictly less than one for $(\alpha > 0, \gamma > 0)$.

6. MODELING MORE COMPLEX USER AND SYSTEM BEHAVIOR

In this section we consider three additional features: (i) user bias towards selecting popular versions; (ii) user local black-



(a) 10,000 polluted versions

(b) Single polluted version

Figure 11. *Bias towards popular versions.* In both cases, total number of polluted copies is 10,000.

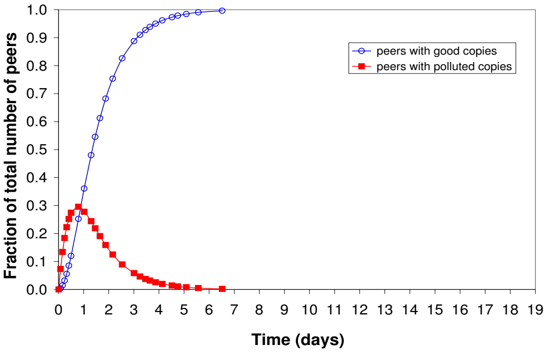


Figure 12. *Blacklisting: Single polluted and single good version*

listing of known polluted versions; and (*iii*) non-negligible download times.

6.1 Bias For Popular Versions

Using the notation in section 2, we now model the probability of selecting a particular version $v \in \mathcal{V}(t)$ as:

$$p_v(t) = \frac{1}{\theta(t)} \left[\frac{n_v(t)}{\sum_{u \in \mathcal{V}(t)} n_u(t)} \right]^{\beta}$$

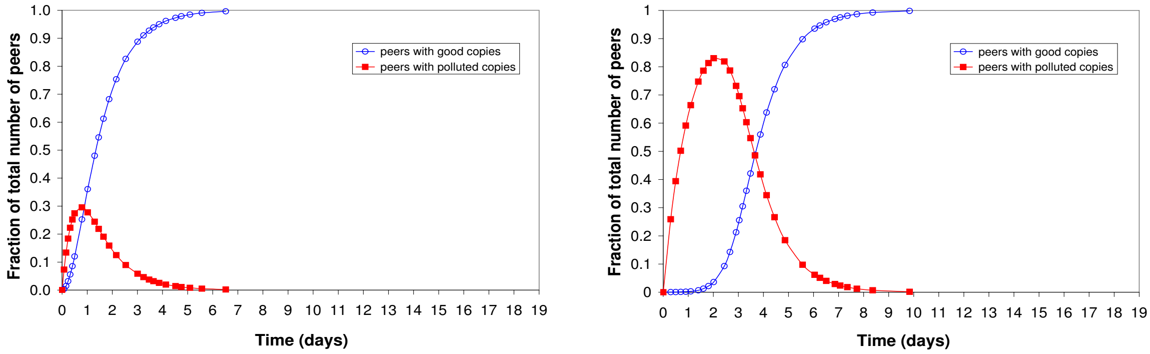
where $\theta(t)$ is a normalization factor and $\beta \geq 0$ controls the degree of bias towards versions with a larger fraction of copies. It is easy to see that $\beta \to \infty$ will imply a version selection behavior wherein the peer selects the most popular version with a probability 1. Also, note that $\beta = 1$ yields the copy-centric model and $\beta = 0$ yields the version-centric model.

The motivation for modeling the selection rule as shown above is as follows. Typically, popular versions (versions with relatively more copies) show up earlier in the search results and also offer higher aggregate download bandwidths through parallel downloading. Depending on the application and the graphical interface, users may be highly biased in their preference to download the more popular versions. This preference may not be completely captured by a value of $\beta = 1$ in the Copy Centric Model. Thus, we study pollution proliferation with slightly higher value of β and present insights on implications for better attack strategies. We can also develop fluid equations for this model and solve them numerically.

We choose a value of $\beta = 1.1$. Through a discrete event simulation of the P2P system we show the resulting proliferation in case of a pollution attack in Figure 11(a) with $N = 10,000$ polluted versions each with one copy, and one good version in the system. Note that with a single good version with $x(0) = 20$ initial copies and each polluted version having a single copy initially, the good version is more popular than any other polluted version. On comparing the Figure 4(b) ($\beta = 1$) and Figure 11(a) ($\beta = 1.1$), we notice that with $\beta = 1.1$ both the extent of pollution and the time for every peer to get a good copy has noticeably decreased. This is because with $\beta = 1.1$ the less popular versions (the polluted versions) are penalized more, resulting in decreased pollution levels. Also, it is worth noticing that with $\beta = 1.1$ even though all peers get a good copy quicker, the number of peers with polluted copies also increases to its maximum (which is less than the maximum with $\beta = 1$) at a faster rate and then decreases to a low value earlier as well. These trends get stronger at still higher values of β. Therefore a better attack strategy would be to increase the relative popularity of polluted versions by having fewer polluted versions spread out over the 10,000 polluted copies. Figure 11 shows the case of a single polluted version with 10,000 copies. In this case, pollution is much more widespread. In light of these observations, we conclude that for the attacker's goal of maximizing the probability of selecting a polluted version, for a fixed number of polluted copies, it is better to have fewer versions with many copies each than may versions with a few copies each. It is surprising that an order of 25,000 polluted versions were discovered for some titles in the FastTrack network [10]. We next explain an attacker's need for increasing the number of polluted versions.

6.2 Blacklisting versions

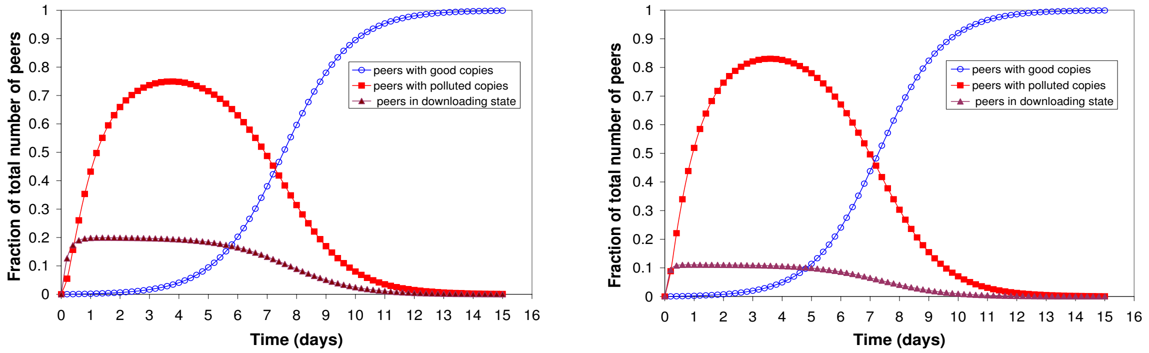
Given how easy it is to attack many P2P systems with pollution, robust P2P systems will need countermeasures. One natural and simple counter measure is the local blacklisting of polluted versions. Here, each peer blacklists all previously downloaded versions that it found to be polluted. One a peer locally blacklists a version, it never downloads that version again. An even more effective counter measure would be to share these local blacklists with other peers, thus resulting in a "global blacklisting" strategy. However complex trust issues and network wide blacklist dissemination issues arise in such a global blacklisting strategy. For example, polluters could themselves start injecting polluted blacklists and in the process blacklist some of the good versions. We show



(a) Single polluted and single good version

(b) 5 Polluted and single good version

Figure 13. *Blacklisting with varying number of polluted versions*



(a) $\eta = 4$

(b) $\eta = 8$

Figure 14. *Proliferation for finite download rates*

that a local blacklisting strategy can go a long way in countering pollution. Blacklisting is less amenable to fluid modeling. The results of this subsection are based on discrete-event simulation.

Figure 13(a) shows the effect of blacklisting for the case of single polluted and single good version with a bias factor $\beta = 1.1$. Notice how effective local blacklisting is for this case. Figure 11(b) is the original proliferation curve without blacklisting and with the same $\beta = 1.1$. The major reason for blacklisting's effectiveness is that it virtually guarantees that each peer will have a good copy in a maximum of two requests. This is because if on its first request the peer downloads a copy of the polluted version, then it will blacklist it before sending out the next request, for which it is bound to select the good version.

For this reason, attackers will find it attractive to introduce many polluted versions into the system so as to thwart the effectiveness of anti-pollution strategies and increase probability of selection of a polluted version. As a next step in Figure 13(b), we plot pollution proliferation with blacklisting for the case when the polluted copies are instead spread out over 5 versions (each version having 2000 copies). Note that local blacklisting is relatively less effective against a larger number of polluted versions available in the system. This is not surprising as with more polluted versions, peers on an average have more versions to blacklist before they download the good version. Thus the attacker would like to increase the polluted versions on offer to decrease the strength of anti-pollution schemes like blacklisting. However, if the attacker spreads out its capacity over too many polluted versions, then the overall effectiveness of the pollution attack may be compromised due to user bias towards the more popular good versions.

6.3 Modeling Non-zero download delay

We now generalize our basic fluid model to account for non-negligible download times for files in the network. Let η be the download rate of a file, so that the average download delay is $1/\eta$. Figure 15 presents the state transition diagram incorporating non-zero download delay. If a peer is in the download state, the peer is in the process of downloading the file. Let $d(t)$ represent the number of peers in the download state and define $x(t)$, $y(t)$, and $p(t)$ as before. The fluid equations for the Copy Centric Model become:

$$\begin{aligned}
\dot{d}(t) &= [M - x(t) - y(t) - d(t)]\mu + \mu y(t) \\
\dot{y}(t) &= \eta d(t)p(t) - \mu y(t) \\
\dot{x}(t) &= \eta d(t)(1 - p(t))
\end{aligned}$$

We reproduce the pollution proliferation results shown in Figure 4(b)($\eta = \infty$) with new $\eta = 4$ and $\eta = 8$ in Figure 14. One can interpret the results of Figure 4(b) as the case with $\eta = \infty$ and deduce that as η decreases the fraction of peers in the downloading state increases. When one compares Figure 4(b) and Figure 14, on the outset it may appear that the pollution level decreases with a decreasing η. However, it turns out that the relative fraction of peers with polluted copies does not significantly change with η. This is shown with a plot of $y(t)/(x(t) + y(t))$ for three different values of $\eta = 4$, $\eta = 8$ and $\eta = \infty$ while keeping the same initial conditions as shown in Figure 4(b). Thus, the increase in the number of peers in downloading state can be equally

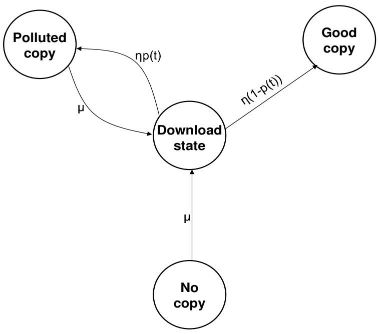


Figure 15. *State transitions for finite download rates*

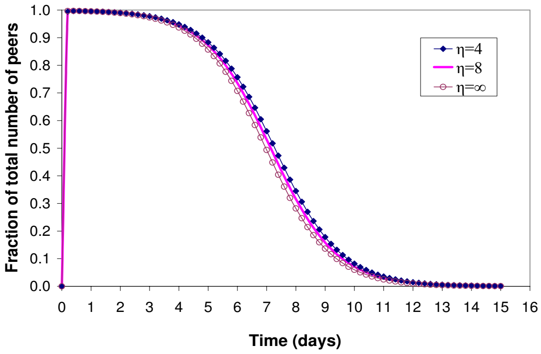


Figure 16. *Change in ratio of polluted copies to total copies with time at different download rates*

attributed to the pool of peers with polluted copies and the pool of peers with good copies.

7. CONCLUSION

We show that it is significantly advantageous for the attacker to launch the pollution attack in advance of the release date of the title. We consider real-world client behavior, including abandoning downloads after too many polluted copies are received and freeloading. We observe that even if a small number of peers abandon downloads or freeload, the number of peers with good copies converges to sub-optimal levels. We also discover that a better attack strategy is to increase the number of polluted versions in order to decrease the effectiveness of anti-pollution schemes like blacklisting. However, if the attacker spreads out its capacity over too many polluted versions, then the overall effectiveness of the pollution attack decreases due to high user bias towards popular versions.

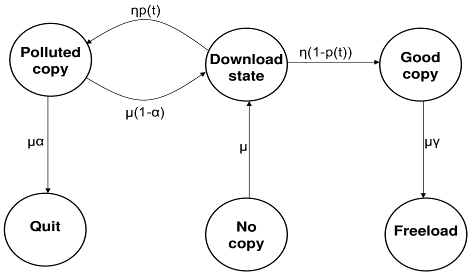


Figure 17. *Modeling peer abandonment, freeloading and non-negligible download times*

References

[1] E. Adar and B.A. Huberman, *"Free Riding on Gnutella,"* First Monday, 5(10), October 2000, www.firstmonday.dk/issues/issue5_10/adar/

[2] http://www.cachelogic.com

[3] B. Cohen, *"Incentives Build Robustness in BitTorrent",* Workshop on Economics of Peer-to-Peer Systems, Berkeley , CA, June 2003.

[4] N. Christin, A. Weigend, and J. Chuang, *"Content availability, Pollution and Poisoning in Peer-to-Peer File Sharing networks,"* Proceedings of the 6th ACM conference on Electronic commerce, Vancouver, Canada, 2005.

[5] F. Dabek, M. Frans Kaashoek, D. Karger, R. Morris and I. Stoica, *"Wide-area Cooperative Storage with CFS,"* Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01), Chateau Lake Louise, Banff, Canada, October 2001.

[6] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, and W. Zwaenepoel, *"Denial-of-Service Resilience in Peer-to-Peer File-Sharing Systems,"* Proceedings of ACM SIGMETRICS 2005, Banff, Canada, June 2005.

[7] H. Feng, A. Kamra, V. Misra and A. Keromytis, *"The Effect of DNS Delays on Worm Propagation in an IPv6 Internet,"* Proceedings of IEEE INFOCOM 2005, March 2005, Miami, FL, USA.

[8] S. Iyer, A. Rowstron and P. Druschel, *"SQUIRREL: A Decentralized, Peer-to-Peer Web Cache,"* 12th ACM Symposium on Principles of Distributed Computing (PODC 2002), Monterey, California, USA, July 2002.

[9] J. Liang, N. Naoumov, and K.W. Ross, *"The Index Poisoning Attack in P2P File-Sharing Systems,"* To appear in proceedings of IEEE Infocom 2006, April 2006, Barcelona, Spain.

[10] J. Liang, R. Kumar, Y. Xi and K.W. Ross, *"Pollution in P2P File Sharing Systems,"* Proceedings of IEEE Infocom 2005, March 2005, Miami, FL, USA.

[11] J. Liang, R. Kumar and K.W. Ross, *"The Kazaa Overlay: A Measurement Study,"* Computer Networks (Special Issue on Overlay Distribution Structures and their Applications), to appear.

[12] http://www.overpeer.com/

[13] A. Rowstron and P. Druschel, *"Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems,"* Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.

[14] S. Saroiu, P. K. Gummadi, and S. D. Gribble, *"A Measurement Study of Peer-to-Peer File Sharing Systems,"* Proceedings of Multimedia Computing and Networking (MMCN'02), San Jose, CA, USA, January 2002.

[15] C. C. Zou, D. Towsley, and W. Gong, *"Email Virus Propagation Modeling and Analysis,"* Umass ECE Dept., Tech. Rep. TR-03-CSE-04, May 2003.