Analysis of the effect of InfoRanking on content pollution in P2P systems

P. Zhang[1], N. Fotiou[2], B. E. Helvik[1], G. F. Marias[2] and G. C. Ployzos[2]

[1]Centre for Quantifiable Quality of Service in Communication Systems(Q2S), Norwegian University of Science and Technology (NTNU) {peiqing.zhang, bjarne}@q2s.ntnu.no

[2]Mobile Multimedia Laboratory, Department of Informatics, Athens University of Economics and Business {fotiou, marias, polyzos }@aueb.gr

January 2014

Abstract

Content pollution is one of the most common attacks against P2P file sharing systems. As such systems are usually open to users and the deployed security mechanisms merely examine the sanity of the downloaded files, content pollution attacks can be easily launched. InfoRanking is a mechanism that tries to mitigate this security risk by ranking content items. In this paper we show through analysis, fluid modeling and simulation that when InfoRanking is used, attackers can deceive users only when they share corrupted copies of legitimate file versions. Nevertheless as corrupted files can be immediately detected after being downloaded this attack is only effective when users enter the system at very low rate and leave relatively fast.

InfoRanking; content pollution; P2P; fluid model; game theory

1 Introduction

P2P file sharing systems generate, even today, a big portion of the Internet traffic. Nevertheless, being used as well for illegally exchanging intellectually property products, such networks create a big income loss to the content industry. In order to inhibit the unauthorized distribution of content, content owners, use both legal means and "attack" methods against such systems. In 2001, Napster, one of the first P2P content delivery systems, was shut down by court order. Moreover various companies, such as Viralg, RetSpan and OverPeer, have been established to protect content from non-authorized distribution on P2P systems by means of *content pollution* [1, 2, 3, 4].

Content pollution is one of the most common attack methods in P2P content delivery networks. By content pollution we mean the sharing of fake or corrupted content instead of the original one. Because in P2P systems everyone is allowed to share content, content pollution attacks can be easily launched. Moreover due to the lack of a central authority content pollution can be hardly prevented. In 2005, it was detected that more than half of the file copies in the KaZaA network were polluted [3].

To counteract content pollution in P2P systems reputation schemes [5, 6, 7, 8, 9, 10] have been proposed. The fundamental principles of these schemes are common, i.e., to predict which user, with high probability, will offer appropriate service. This prediction is usually based on user past behavior.

All these methods usually demand modifications of the already deployed protocols. Moreover they have to give users incentives in order to vote correctly. Fotiou et al. proposed *Infornaking* [11], a light-weight solution that excludes pollution while relaxing the above requirements. InfoRanking is based on the observation that in P2P systems malicious users share more versions of the same content than legitimate users [3, 12]. Moreover it considers positive votes only, which makes its implementation easier as users' actions, such as the fact that they are sharing a file after downloading it, can be regarded as an implicit positive vote.

This paper gives a formal analysis on the performance of InfoRanking and shows how it does limit the effect of attackers' behavior, helping at the same time P2P systems to improve their long-term performance. This paper is organized as follows. Section 2 presents the content pollution attack and introduces InfoRanking. Section 3 presents related work in the area. In Section 4, our methodology is presented and Section 5 gives some basic analysis about the game between attackers and users. Deeper analysis of attacker and user behavior is made in Sections 5.2 and 5.1 respectively. In Section 6 fluid models are built to measure the performance of InfoRanking. Finally, Section 7 presents our conclusion and future work plans.

2 Background

2.1 Content Pollution

Users in P2P systems initially search for the piece of content they want to download using keywords via the P2P application interface. The P2P application may return thousands of results matching these keywords, especially if the content is popular. Among these results, there will be different versions of the same content. For example the song "The Scientist" may have a mp3 version, a wma version, a version performed by the band "ColdPlay", a version performed by some unknown singer etc. Versions are distinguished from each other by their "metadata" (e.g., file name, file extension, file properties, keywords etc.). Versions of which the metadata match their actual content are referred to as *clean versions*, otherwise they are called *fake versions*. Example of fake versions can be an executable file masked as a video file, or a file of a song performed by an artist A, which metadata denotes that it is a song of artist B.

Each version has multiple copies. All copies of a version are expected to have the same–well known–hash value[1]. A file advertised as a copy of a version is a *clean copy* if its computed hash value matches the expected hash value, otherwise it is a *corrupted copy*. Figure 1 depicts the above concept.

Detection of polluted items, i.e., fake copies and corrupted versions, is a two step process. The first step is performed automatically, immediately after a file has been downloaded; in this step it is checked whether the file is a corrupted copy or not. The second step involves user interaction and it is performed some time after the file has been downloaded. During the second step it is checked whether the downloaded file is a fake version. Corrupted copies are automatically deleted immediately after being downloaded, whereas fake versions are deleted by the user and only after he has detected them. Fake versions are shared by the users that have downloaded them (i.e. are made available to others), until either they delete them (e.g., because they discover that they are fake versions), or the users leave the system.

2.2 InfoRanking

InfoRanking is a vote-based approach for ranking information items in user-driven information distribution networks. InfoRanking is based on the observation that in those networks, malicious users provide numerous fake versions of the same information item in order to avoid blacklisting [3, 12]. When InfoRanking is used, users may vote only positively regarding a specific information item. Moreover a user may vote only once.

When it comes to a P2P file sharing network, users may vote for a file. A user's vote for a specific file shows that the user believes that this is a "good" file. The meaning of "good" depends on the con-

[1] it can be learned for example through a tracker

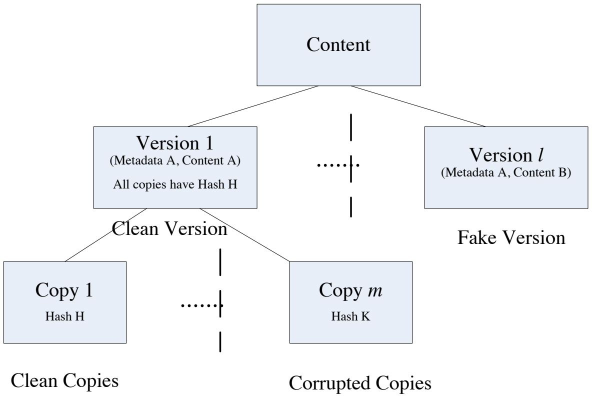

Figure 1: Illustration of terms clean version, clean copy, fake version and corrupted copy

Versions	**Users**	**Score**
The Free movie HQ.avi	U1, U2, U3, U4	0.25 + 0.5 + 0.5 + 0.5 = 1.75
The Free movie DVDRip.avi	U1, U2, U3, U4	0.25 + 0.5 + 0.5 + 0.5 = 1.75
The Free movie Xvid.avi	U1, U5, U6, U7	0.25 + 1 + 1 + 1 = 3.25
The Free movie TOM Rip.avi	U1	0.25

Table 1: InfoRanking voting example

text under which InfoRanking is used. In this paper we consider as "good" files the unpolluted files. Because in InfoRanking users vote only positively, the fact that a user shares a file can be considered as a vote. Therefore, not only there is no need for the deployment of a separate voting subsystem, but as long as a user shares a file he participates in the voting procedure, so voting incentives are unnecessary.

In a subset V of all files in the system–such as a list of files matching a keywords based search–the score of each file is the sum of all the weighted votes in that subset. Each vote of a user U in V is weighted by a factor w computed as $w = 1/(\sum U_C)^a$ where $\sum U_C$ is the sum of U's votes in V and a is a fixed value. As an example consider a P2P file-sharing network where a user searches for the movie "The free movie". He receives the 4 results shown in Table 1. The InfoRanking based score is calculated using $a = 1$. The first column of Table 1 contains all the versions that are included in the result set. The second column contains a list of users that share each version and the third column contains the score of each version. As it can be seen in this table user U1 shares 4 versions in the result set, therefore he has "voted" 4 times and his vote is weighted by 0.25. On the other hand users U5, U6, and U7 have voted only once so their votes are weighted by 1. The rank of each file is calculated by summing the weighted votes. In this example the version "The Free movie Xvid.avi" has the highest score therefore this is the one that will be chosen by the user.

The score of each version is computed in a distributed way. Upon receiving a result set each user calculates the score of each version using the aforementioned formula. Each user may also maintain a blacklist of versions and users; a user will never download a version contained in the blacklist and will never consider the vote of a blacklisted user.

It can been seen that when InfoRanking is used, in order for malicious users to achieve a successful attack, they should outnumber benign users. In a result set in which each malicious user shares on average F_m versions, each benign user shares on average F_g versions and there exist U_g benign users, the number of malicious users should be $U_m > (F_m/F_g)^a * U_g$ in order to lead to the selection of a fake version.

By ranking the versions of content rather than the users, InfoRanking is more robust to Sybil attacks than a user-ranking system. Nevertheless there can be cases in which the number of malicious users is so big that a benign user may be convinced to download a polluted item. In those ultra-polluted networks centralized black lists of polluted items can be used. Those black lists will force malicious users to share even more closely related files in order to achieve their attack and therefore to lower their vote's weight.

Finally by allowing positive votes only and by considering the fact that user shares a file as a positive vote, there is no way for attackers to negatively affect the rank of a file.

3 Related Work

Our paper's contribution is twofold. We show the efficiency of a content pollution prevention mechanisms and we present an analytical approach for modeling P2P systems combining game theory and fluid models. Therefore, related work in two domains is examined; work related to content pollution prevention and work related to analytical evaluation of the impact of attacks in a P2P system.

3.1 Content Pollution Prevention

Content pollution mechanisms can be distinguished to those that rank users and those that rank content. User ranking mechanisms–such as EigenTrust [8] and Scrubber [13]–deploy voting schemes which allow each user to rank others based on their behavior. The voting results are used in order to build trust relationships. These relationships are propagated, leading to the creation of chains of trust. The rank of each user is calculated in a distributed manner, i.e., each user calculates his own personal rank about the other users. User ranking approaches suffer from two basic drawbacks (i) a newcomer does not know who to trust unless his application is pre-configured with a list of trusted users and (ii) usually it is easy for a user in a P2P system to change his identity, therefore to "reset" his rank. InfoRanking overcomes these problems; when a user receives his query results, no matter if he is a newcomer or not, he is able to calculate the rank of each item. Moreover, the identity of each item–usually the result of a hash function applied over its data–is constant and unchangeable.

In contrast to user ranking approaches, content ranking solutions rank items. Credence [14] is a typical scheme of this type. Credence is a weighted voting protocol in which a user may vote positively or negatively on any object regarding its authenticity. Any user wishing to download some content issues a vote-gather query to collect votes on candidate objects. This query is flooded to the network and each user that posses votes, responds. Credence–in contrast to InfoRanking–requires the modification of the application protocol in use. Moreover Credence's vote gathering procedure adds a communication overhead.

Hybrid solutions–such as a modified version of Scrubber presented in [15]–try to get the best of both worlds. Nevertheless they introduce significant complexity.

In general voting schemes require users' cooperation, i.e., users should be willing to vote for other users–or items–and share their votes with others. As a result voting schemes do not have only to cope with users not voting correctly but also with creating mechanisms that give incentives to users to vote–such as the one proposed in [16]. However InfoRanking does not face this problem. By using positive votes only, regular users' actions can be considered as votes, for example the fact that a user is sharing a file can be considered as a positive vote for this file. As we show below, this approach leads to very competitive results.

3.2 Analytical Evaluation

Few research efforts study the impact of content pollution in P2P systems using an analytical approach. Kumar et al. [17] are using fluid models to model pollution proliferation in P2P systems. Nevertheless, they do not consider any security solution. Lee et al. [18] create a mathematical model to assess the impact of pollution on file popularity evolution by studying human behavior. Their research is only focused on attackers that pollute a P2P system with polluted versions and it does not consider any security mechanism. Our work considers attackers that pollute a P2P system with either fake versions or fake copies. Analytical modeling has also been used to study content pollution impact in P2P live streaming systems–such as in [19]. However these models can not be applied in P2P file sharing systems, because in live streaming chunks are not retransmitted, therefore in case a user receives a corrupted chunk he can not re-download it again.

Finally, game theory has been used in various research efforts to study the free-riding effect in P2P systems [20, 21, 22]. Our work is focused on another type of attack: content pollution.

4 Methodology

In P2P systems, users' goal is to download a specific piece of content, such as a software package, a movie, or a song. On the other hand, attackers try to prevent users from achieve this by inserting corrupted copies in the system. Obviously, users and attackers have conflicting interests; users want to increase the probability of downloading a clean copy, whereas attackers want to prevent this. In the real world, these conflicts are usually caused by copyright arguments and intended added advertisements, i.e., economic benefit associated with the content and its popularity; popular content has much higher commercial value than unpopular one. In this paper popular content is cosidered, i.e., content that attracts more than thousands of downloaders. Moreover the following assumptions are made:

- A1: Compared to the large number of downloaders, the number of attackers is limited.
- A2: Users intend to minimize their probability of receiving and keeping/sharing polluted items.
- A3: Attackers try to maximize the probability of users receiving polluted items.
- A4: Among all the strategies leading to the same pollution effect, attackers will choose the one with the lowest cost.
- A5: Each benign user only shares a single version of a content item, while attackers usually share multiple versions. Both benign users and attackers share only one copy for each version.
- A6: From the downloader's perspective, the copies of the same version look identical. Therefore, no matter whether they use InfoRanking or not, users select a copy to download at random (i.e., uniformly distributed across all the copies of the selected version)

Based on those assumptions, we use three approaches to analyze the effects of InfoRanking: (i) behavior predication inspired by game theory (ii) system modeling using fluid models and (iii) simulation.

Initially the strategies that users and attackers can follow are examined and the payoff of each strategy with and without InfoRanking is calculated. As a next step fluid models are used to abstract the evolution of the whole system and to analyze it in steady state, i.e., a state in which the rate of users entering the systmes and leaving are almost equal. These analytical results are validated using the OMNet++ discrete event simulator [23]. In our analysis, we are considering that versions are ranked based on their InfoRanking scores. If two or more versions have the same score, then they are ranked randomly–with respect to each other.

In the remainder of this paper the following notation describing the system is used:

- V: The set of all versions of a content item in the system.
- $|V|$: The number of versions in the system.
- V_c: The subset of the set V which includes all the clean versions in the system.
- V_p: The subset of set V which includes all the fake versions in the system, $V_c \bigcup V_p = V$, $V_c \bigcap V_p = \emptyset$.
- $|V_c|, |V_p|$: The number of version in V_c and in V_p respectively.
- $v^i \in V$: The ith ranked version.
- v_c^i: The set of all clean copies of v^i.
- v_p^i: The set of all corrupted copies of v^i., $v_c^i \bigcup v_p^i = V$, $v_c^i \bigcap v_p^i = \emptyset$.
- $|v|$: The number of copies of version v.
- Score^i: The score of a version i, calculated using InfoRanking.

- $\hat{s}$ an attack strategy.
- $\mathsf{Score}^i(\hat{s})$ the score of a version i after the attacker strategy $\hat{s}$.
- $\mathsf{P}(\hat{s})$ the probability that a user selects a corrupted copy under attack strategy $\hat{s}$.

5 A Game between Users and Attackers

The pollution problem can be considered as a battle between conforming users and attackers. In this battle users and attackers can follow various strategies.

A user's strategy for selecting a version to download depends on whether an object reputation scheme exists or not. If there is not any scheme the user will simply choose an available version randomly, or according to his own habits. When InfoRanking is used, the user has the possibility to choose a version according to the corresponding scores given by InfoRanking. Therefore, the possible user strategies are:

1. Select an available version at random.
2. Select the version with the highest score. If there are several versions with the same score, select a version uniformly randomly among them.

If a user downloads a corrupted copy of the version with highest score, there are two strategies for him to chose as a next step:

1. Select another copy from this version.
2. Select a copy from another version.

An attacker can choose among several strategies. In general, these strategies are composed by two basic actions: adding a corrupted copy to an existing version or inserting a new fake version to the system. A strategy is denoted as: $\hat{s}^r$. If $r \leq |V|$ then the attacker adds a corrupted copy to the rth ranked version, i.e, v^r, else the attacker adds a new fake version into the system.

Attackers do not have any incentive to add corrupted copies of the fake versions, as the corrupted copies will be immediately detected, therefore the fake versions will not spread.

5.1 User Strategy

In this section, user behavior is analyzed and it is shown that:

- Always selecting copy from the version with highest score is the best strategy for users
- If a user obtains a corrupted copy then the best strategy for him is to select another copy from this version.

We consider a network in which attackers follow a greedy strategy (denoted as $\hat{s_g}$). They add as many as possible fake versions and share a corrupted copy for each version in V_c. Therefore, if there are N independent attackers in the system, each of whom is able to insert k fake versions, for every clean version there are N corrupted copies and the total number of polluted items (corrupted copies of clean versions and fake versions)[2] is $N \cdot k + N \cdot |V_c|$

5.1.1 Best Strategy for New Users

If users randomly select a version from the set V and if there are N independent attackers in the system, each of whom is able to insert k fake versions, then the probability $P(\hat{s_g})$ that a new coming user will select and download polluted content equals:

$$\mathsf{P}(\hat{s_g}) = \frac{N \cdot k + N \cdot |V_c|}{N \cdot k + N \cdot |V_c| + \sum_{j \in \{1 \ldots |V_c|\}} |v_c^j|}$$

Since v^1 is the clean version with the largest number of clean copies, from the user's perspective, $\sum_{j \in \{1 \ldots |V_c|\}} |v_c^j| \leq |V_c| \cdot |v_c^1|$, i.e., all clean versions have at most many copies as v^1 [3]. Therefore:

[2] According to assumption A5 both benign users and attackers share only one copy of each version

[3] Actually, v^1 denotes the highest ranked version. However, according to the analysis presented later in 5.2, the clean version initially with the largest number of clean copies will always be the highest ranked one and with the largest number of clean copies.

$$\mathsf{P}(\hat{s_g}) \geq \frac{N \cdot k + N \cdot |V_c|}{N \cdot k + N \cdot |V_c| + |V_c| \cdot |v_c^1|} \geq \frac{N}{N + |v_c^1|}$$

However $\frac{N}{N+|v_c^1|}$ is the probability that a user will download polluted content if he randomly selects to download a copy from the highest ranked version. Therefore selecting a copy form the highest ranked version always has the smallest probability of pollution.

5.1.2 Best Strategy for Users Experiencing Pollution

If a user, who has obtained a corrupted copy from the version with highest rank, selects another version to download, then

$$\mathsf{P}(\hat{s_g}) \geq \frac{N \cdot k + N \cdot (|V_c| - 1)}{N \cdot k + N \cdot (|V_c| - 1) + (|V_c| - 1) \cdot |v_c^1|}$$

$$> \frac{N}{N + |v_c^1|} > \frac{N - 1}{N + |v_c^1| - 1}$$

However $\frac{N-1}{N+|v_c^1|-1}$ is the probability of pollution if the user selects another copy of the highest ranked version. Therefore in case of pollution the best user strategy is to select another copy from the highest ranked version, assuming a greedy attacker.

5.2 Attackers' Strategies

In order to analyze attackers' strategies we initially consider the case of the first coming attacker and then we generalize it to the case of multiple independent attackers. In this section we assume that users always select the highest ranked version.

5.2.1 Strategy of the First Coming Attacker

Let A_1 be the first attacker trying to pollute a system in which there already exist l clean versions i.e., $|V| = |V_c| = l$. Two cases are considered; the case in which $|v^1| > 1$, i.e., the are more than one copies of the highest rank version, and the case in which $|v^1| = 1$, i.e., there is a single copy of the highest rank version, and we will show that the best attack strategy for the attacker A_1 in both cases is the same; $\hat{s}^1$ i.e., to insert a corrupted copy to the clean version with the highest rank.

Case $|v^1| > 1$

Obviously, if initially $|v^1| = |v_c^1| > 1$, it is meaningless for an attacker to add fake versions. The maximum score of a newly added version is 1, which is strictly less than the score of the highest ranked version[4]. Therefore the fake version will never be selected by users. In order to achieve any disturbance to the system, this attacker has to add corrupted copies to the version with the highest rank. Assume that v^j is the last ordered item in V which satisfies that $|v^j| = |v^1|$. If $j = l$, namely all clean versions have the same number of clean copies, any strategy $\hat{s}$ leads to the same $\mathsf{P}(\hat{s}) = 1/(|v^1| + 1)$. In that case the attack strategy with lowest cost is to insert a corrupted copy to one version i form V. Then this version will have the highest score and should be ordered as the first item in V.

In case $j < l$ and $|v^j| - |v^{j+1}| > 1$, adding a corrupted copy in version $j+1$ to version l has no effect, as $\mathsf{Score}^i(\hat{s}), i \in \{j+1, ..., l\}$ will be less than any $\mathsf{Score}^i(\hat{s}), i \in \{1, ..., j\}$. Thus, A_1 has to add corrupted copies to the top j versions. Similarly from the attacker's perspective the strategy with the best effect and the lowest cost is to add a corrupted copy to a single version which number of copies equal to $|v^1|$.

In case $j < l$ and $|v^j| - |v^{j+1}| = 1$, adding a corrupted copy to a version which number of copies equals to $|v^1|$ will result to:

$$\mathsf{P}(\hat{s}^1) = \frac{1}{|v^1| + 1}$$

On the other hand adding a corrupted copy to a version which number of copies equals to $|v^{j+1}|$ will result to:

$$\mathsf{P}(\hat{s}^{j+1}) = \frac{1}{j+1} * \frac{1}{|v^{j+1}| + 1} \geq 0.5 * \frac{1}{|v^1|}$$

[4] Since each benign user only shares a single version of a content item–according to A:5–the score of each version before any attacker enters the system equals to the number of copies of this specific version.

For every $|v^1| > 1$, the equation $\mathsf{P}(\hat{s}^1) > \mathsf{P}(\hat{s}^j + 1)$ always stands.

In a nutshell if initially $|v^1| = |v_c^1| > 1$, the best strategy for the first attacker A_1 is always $\hat{s}^1$, i.e. adding one corrupted copy to the highest ranked version.

Case $|v^1| = 1$

The case in which initially $|v^1| = |v_c^1| = 1$, i.e., for every clean version there is only one clean copy in the system, is now examined. When A_1 enters the system, he can follow two attack strategies; $\hat{s}^1$ and $\hat{s}^{l+1}$, i.e., add a corrupted copy to the highest ranked version or add a new fake version. The impact of each strategy differs when compared in short-term time frame and in a long-term one. The probability of pollution in short time frame is:

$$\mathsf{P}(\hat{s}^1) = 1/2$$

$$\mathsf{P}(\hat{s}^{l+1}) = 1/(l+1)$$

As we always have $\mathsf{P}(\hat{s}^1) \geq \mathsf{P}(\hat{s}^{l+1})$, in short term time frame the best strategy is $\hat{s}^1$.

The long term effect can not be calculated in a straightforward way, as the evolution of the system has to be studied. Since fake versions can only be identified after the content is previewed by the user, in contrast to corrupted copies which can be detected immediately after being downloaded, $\hat{s}^{l+1}$ in the long term will result in the spread of the pollution. In order to study the strategies' long term impact fluid models are created. The following notation is used:

- λ: The rate at which downloaders complete download.
- ι: The rate at which clean copies leave the system.
- μ: The rate at which users become aware of having downloaded a fake version and delete it.
- ω: The rate at which users abort the download.
- ξ: The rate at which new requesting users arrive.

The strategy $\hat{s}^1$ leads to the simple model of Figure 2 (a). $\mathsf{Score}^1(\hat{s}^1)$ is always greater than the score of any other version in the system. Users may enter only in three states. The first state indicates the procedure of searching and downloading. Since there is only one corrupted copy of $|v^1|$ the user in the first state gets a clean copy, with probability $|v^1|/(|v^1|+1)$, and is transferred to the respective state. With probability $|v^1|/(|v^1|+1)$, he obtains the corrupted copy. In that case, the hash value helps the user to detect the pollution immediately. Since the user experienced the pollution never downloads the same copy again, he will get a clean copy with rate λ in the next step.

The system state can be described by a vector $\pi_1(t) = (m(t), |v^1(t)|, y(t))^T$, where $m(t)$ indicates the number of users at the first sate, $y(t)$ represents the number of users downloaded corrupted copies or fake versions. The evolution of $\pi_1(t)$ can be described by fluid model as follow:

$$\frac{d\pi_1(t)}{dt} = \mathsf{Q}_1(t) \cdot \pi_1(t) + \begin{pmatrix} \xi \\ 0 \\ 0 \end{pmatrix} \tag{1}$$

where the transition matrix $\mathsf{Q}_1(t)$ satisfies

$$\mathsf{Q}_1(t) = \begin{pmatrix} -\lambda - \omega & 0 & 0 \\ \lambda \cdot \frac{|v^1(t)|}{|v^1(t)|+1} & -\iota & \lambda \\ \lambda \cdot \frac{1}{|v^1(t)|+1} & 0 & -\lambda - \omega \end{pmatrix} \tag{2}$$

For the second strategy, $\hat{s}^{l+1}$, we consider the worst case for the user; in the system there is only one clean version, v_{clean} with one copy, i.e. $l = 1$, therefore the attacker's strategy is actually $\hat{s}^2$, i.e., he inserts a fake version, denoted as v_{fake}, in the system. The number of copies of v_{clean} and v_{fake} is a function of time t, namely $|v_{\mathsf{clean}}(t)|$ and $|v_{\mathsf{fake}}(t)|$ respectively. The state the system can be described by the vector $\pi_2(t) = (m(t), |v_{\mathsf{clean}}(t)|, |v_{\mathsf{fake}}(t)|, y(t))^{\mathsf{T}}$.

Let $I[x(t), y(t)]$ be a function of $x(t)$, $y(t)$ which satisfies that:

$$I[x(t), y(t)] = \{1x(t) > y(t)0.5x(t) = y(t)0x(t) < y(t) \tag{3}$$

We assume that the pollution inserted by the attacker is persistent, i.e., one copy of the fake version v_{fake} never leaves the system. The evolution of $\pi_2(t)$

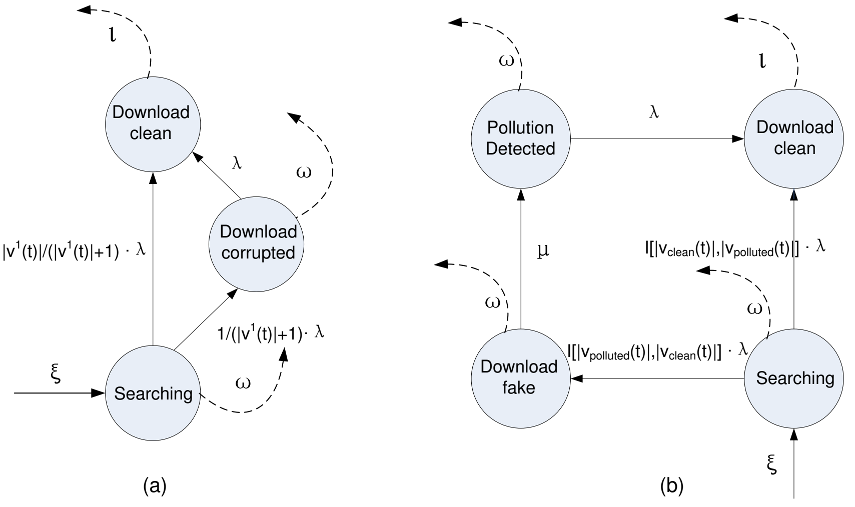

Figure 2: The fluid model to analyze the long-term effect of (a) $\hat{s}^1$ and (b) $\hat{s}^{l+1}$

can be described as follows:

$$\frac{d\pi_2(t)}{dt} = \mathsf{Q}_2(t) \cdot (\pi_2(t) - \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}) + \begin{pmatrix} \xi \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{4}$$

where the transition matrix $\mathsf{Q}_2(t)$ satisfies

$$\mathsf{Q}_2(t) = \begin{pmatrix} -\lambda - \omega & 0 & 0 & 0 \\ \lambda \cdot A & -\iota & 0 & \lambda \\ \lambda \cdot B & 0 & -\omega - \mu & 0 \\ 0 & 0 & \mu & -\omega - \lambda \end{pmatrix} \tag{5}$$

where

$$A = I[|v_{\mathsf{clean}}(t)|, |v_{\mathsf{fake}}(t)|]$$

$$B = I[|v_{\mathsf{fake}}(t)|, |v_{\mathsf{clean}}(t)|]$$

By setting $d\pi_1(t)/dt = 0$ in the equations group (1) and (2) and $d\pi_2(t)/dt = 0$ in the equations group (4) and (5) respectively and solving them, we can calculate the probability that a new coming user experiences pollution under attack strategy $\hat{s}^1$ and $\hat{s}^{l+1}$ when system is in steady state.

If the users that share a clean copy leave the system too soon, the number of clean copies in the system will become 0 in the long term. At that time, new coming users can only get a copy of the fake version. That is not the case we are interested in, we rather focus on the case that the system will not collapse because it lacks clean content. The probability that a user selects a corrupted copy under $\hat{s}^1$ and $\hat{s}^{l+1}$ respectively equals:

$$\mathsf{P}(\hat{s}^1) = \frac{\iota \cdot (\lambda + \omega)}{\xi \cdot \lambda} \tag{6}$$

$$\mathsf{P}(\hat{s}^{l+1}) = \{0.5\mu < K0\mu \geq K \tag{7}$$

where $K = \frac{\iota \cdot (\lambda + \omega) \cdot (\xi \cdot \lambda + \omega \cdot (\lambda + \omega))}{\xi \cdot \lambda^2 - \iota \cdot (\lambda + \omega)^2}$

Thus, in both short term and long term, if users check what they have download within a reasonable time, $\hat{s}^1$ has bigger negative impact to the system than $\hat{s}^{l+1}$. Assuming that users want to maximize their benefits, they will check within reasonable time, i.e. within

$$\frac{\xi \cdot \lambda^2 - \iota \cdot (\lambda + \omega)^2}{\iota \cdot (\lambda + \omega) \cdot (\xi \cdot \lambda + \omega \cdot (\lambda + \omega))}$$

time unit.

To sum it up, in case that initially:

$$|v^1| = |v_c^1| = 1$$

the best strategy for the first attacker A_1 is always $\hat{s}^1 = \{v^1\}$, i.e. adding one corrupted copy to the clean version with the highest score. In the case

$$|v^1| = |v_c^1| > 1$$

then $\hat{s}^1$ is again the best attack strategy

5.2.2 Multiple Independent Attackers

The single attacker case can be expanded to study the effect of multiple independent attackers, i.e., attackers do not co-operate to achieve the maximum negative impact.

When an attacker enters the system, he does not know if there is pollution in the system or not. To maximize the impact of his attack, he adds a corrupted copy to the version with highest score. Furthermore, an attacker can reasonably assume that all the other attackers have taken or will follow the same strategy. Therefore, polluting the version will highest score ensures that, firstly, the corrupted copy inserted will be selected and, secondly, the pollution from all attackers has cumulative impact.

Actually, in P2P system with InfoRanking, a phenomenon similar to Matthew effect can be observed; the version with the highest score and providing that it is a clean version, always stays as the top ranked version.

5.3 Simulation Results

The analytical results obtained from the previous sections are validated in this section using simulation. A BitTorrent-like environment is simulated in which the following roles exist: seeders, i.e., users that bootstrap the system with clean versions, attackers, i.e., users who try to pollute the system, and regular users who are trying to obtain an information item. That of a user–henceforth called simply “users”–is the only role that downloads content. Moreover users is the only role that after some period of time leaves the system. In our set up initially seeders enter the system and they are assigned a number of files to share. Then attackers enter the system and deploy their attack based on certain strategies. Users download items, detect pollution and leave the system stochastically. A global oracle is responsible for collecting the appropriate statistics. All events are created following the Poisson distribution.

Initially we examine the case of a single attacker and two attacking strategies, namely $\hat{s}^1$ and $\hat{s}^{l+1}$. For the $|v^1| = |v_c^1| = 1$ case we consider a network in which a clean copy of a clean version is initially shared by a single seeder. The seeder never leaves the network. Moreover there exists an attacker which can choose between two strategies: $\hat{s}^1$ i.e. the attacker adds a corrupted copy to the clean version v^1 and $\hat{s}^{l+1}$, i.e., the attacker adds a new, fake version into the system.

Figure 3 shows the probability of a user getting polluted under different attack strategies. These results have been obtained by simulating a system in which 5000 users in total enter the network and abandon it with rate (μ). The rest of the simulation parameters are $\lambda = 0.6$, $\iota = 0.2$ and $\xi = 10$.

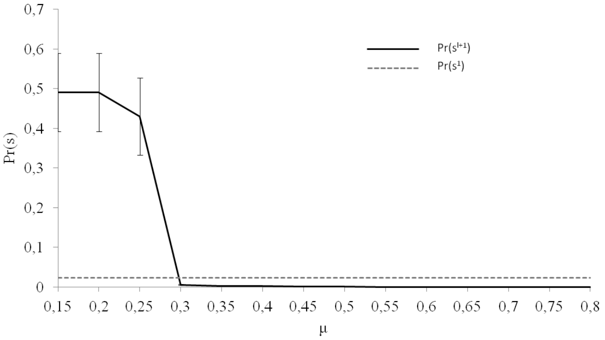

Figure 3: Probability of receiving a corrupted copy with varying μ and different attack strategies

As it can be seen if InfoRanking is used and users check for pollution relatively fast, in the long term the probability of pollution becomes 0 when the $\hat{s}^{l+1}$ attack strategy is followed. In this setup if users check for pollution with rate $\mu > 0.25$, i.e., 2.4 times the time they need to download a file[5], the fake version is not propagated therefore in the long term no user gets polluted.

[5] in this example users complete a download in $A = 1.667$ time units and detect pollution in $B = 4$ time units, $B = 2.4 * A$

Figure 4 shows the probability of pollution when attack strategy $\hat{s}^1$ is followed, with varying ι and ξ with $\lambda = 0.5$ and abandon rate equal to zero. Both simulation and analytical results show that the smaller the rate with which clean copies leave the system, the smaller is the probability of pollution. Similarly the higher the rate users enter the system, the smaller is the probability of pollution. This happens because the number of corrupted copies remains stable, whereas the number of clean copies increases in proportion to the rate that users enter the system and inversely proportional to the rate clean copies leave the system.

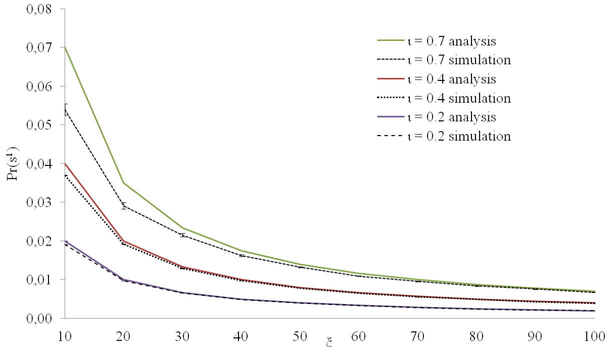

Figure 4: Probability of pollution for varying ι and ξ

The case which $|v^1| > 1$, i.e., when the highest ranked version's score is greater than one, is also simulated. In that case a single attacker can achieve the biggest negative impact with the lowest cost by adding a corrupted copy to the version with the highest score. As it can be seen in Table 3, adding a fake version has no effect on the system due to InfoRanking, while polluting all the versions of a file has the same effect as polluting the best scored version and polluting a random version of the file has very low impact. In this simulation setup we consider a varying number of initial seeders, 8 versions of each file with the versions assigned to seeders using a zipf distribution. Each experiment is repeated 10 times for 500 users . The rest of the simulation parameters are: $\lambda = 0.5$, $\iota = 0.2$, $\mu = 0.8$ and $\xi = 10$ The case of multiple independent attackers is also simulated.

	Number of Seeders		
Attack Strategy	**10**	**20**	**30**
Pollute the best version	0.01815	0.01699	0.01600
Pollute all versions	0.01801	0.01690	0.01588
Add a fake version	0	0	0
Pollute a random version	0.00370	0.00330	0.00470

Table 2: Probability of receiving a polluted item under different attack strategies

Initially we simulate two systems, one being attacked by 10 attackers and one being attacked by 40 attackers. In both cases attackers add a corrupted copy to the version with the highest rank. We run the experiments for varying ι and ξ and by setting $\lambda = 0.5$ and the abandon rate equal to zero.

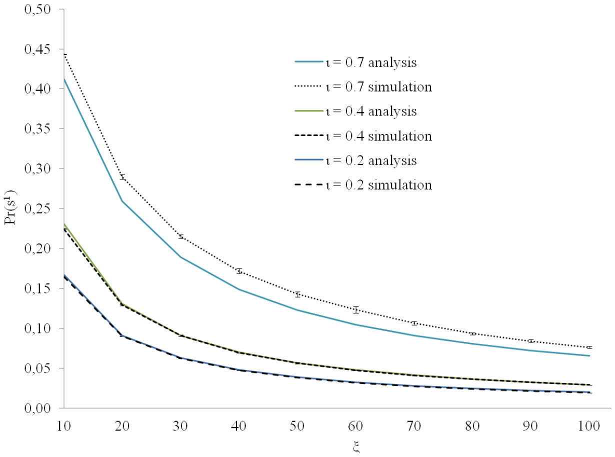

Figure 5: Probability of receiving a corrupted copy with 10 independent attackers

As it can be seen from Figures 5 and 6 both analytical and simulation verify that the probability of pollution depends on the rate users enter and leave

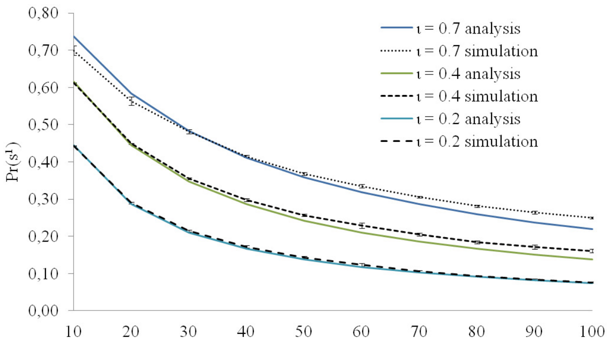

Figure 6: Probability of receiving a corrupted copy with 40 independent attackers

the system. Figure 7 shows the probability of receiving a corrupted copy when the number of attackers varies and $\lambda = 0.5$, $\iota = 0.3$, and $\xi = 50$.

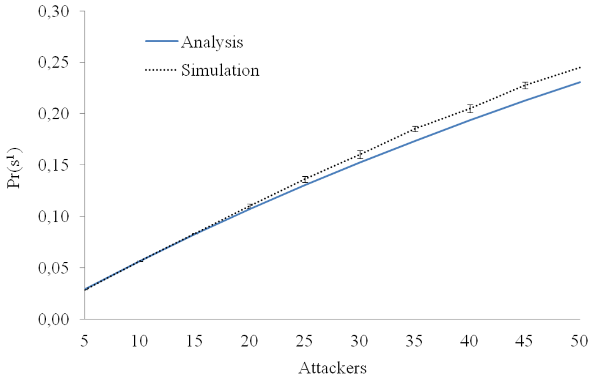

Figure 7: Probability of receiving a corrupted copy with varying independent attackers

6 The Effect of InfoRanking

6.1 Modeling a P2P System with InfoRanking

In the previous section we concluded that in a system where InfoRanking is used:

- Independent attackers achieve the maximum impact by inserting a corrupted copy into the version with highest rank
- Users achieve the maximum gain by selecting a copy from the version with the highest rank, by checking the content the get fast enough and once they get a corrupted copy, by selecting another copy from the same version.
- The version with highest rank, providing that it is a clean version, always remains the top ranked version.

We are now modeling a P2P system in which InfoRanking is applied, using a fluid model. Similarly to the previous section, we are using the vector $\pi(t) = (m(t), |v_c^1(t)|, y(t))^T$ to describe the system, $m(t)$ is the number of users searching or downloading the content, $y(t)$ is the number of users experiencing pollution. Assuming that there are N independent attackers in the system, the evolution of $\pi(t)$ can be described as follows:

$$\frac{d\pi(t)}{dt} = \mathsf{Q}(t) \cdot \pi(t) + \begin{pmatrix} \xi \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

where the transition matrix $\mathsf{Q}(t)$ satisfies:

$$\mathsf{Q}(t) = \begin{pmatrix} -\lambda - \omega & 0 & 0 \\ \lambda \cdot \frac{|v_c^1(t)|}{|v_c^1(t)|+N} & -\iota & \lambda \cdot \frac{|v_c^1(t)|}{|v_c^1(t)|+N} \\ \lambda \cdot \frac{N}{|v_c^1(t)|+N} & 0 & -\lambda \cdot \frac{|v_c^1(t)|}{|v_c^1(t)|+N} - \omega \end{pmatrix} \tag{9}$$

To study the system in steady-state, we set the left part of equation (8) equal to zero, then by solving equation (9), the expected number of users in each state can be calculated as follows:

$$E(m) = \frac{\xi}{\lambda + \omega} E(|v|_c^1) = \frac{\xi \cdot \lambda - N \cdot \iota \cdot \omega}{\iota \cdot (\lambda + \omega)} E(y) = \frac{N \cdot \iota}{\lambda + \omega} \tag{10}$$

The expected probability that a coming user selects a corrupted copy is:

$$E(P) = \frac{N}{E(|v|_c^1) + N} = \frac{N \cdot \iota \cdot (\lambda + \omega)}{\lambda \cdot (\xi + N \cdot \iota)} \tag{11}$$

From equation (10), we also know that the expected total number of users remaining in the system equals:

$$\begin{aligned} &E(m) + E(|v|_c^1) + E(y) = \\ &= \frac{\xi \cdot (\lambda + \iota) + N \cdot \iota \cdot (\iota - \omega)}{\iota \cdot (\lambda + \omega)} \end{aligned} \tag{12}$$

According to Little's Formula [24], in a stable P2P system with InfoRanking, the mean time of a user in the system is:

$$\begin{aligned} E(T) &= \frac{E(m) + E(|v|_c^1) + E(y)}{\xi} = \\ &= \frac{\xi \cdot (\lambda + \iota) + N \cdot \iota \cdot (\iota - \omega)}{\xi \cdot \iota \cdot (\lambda + \omega)} \end{aligned} \tag{13}$$

6.2 Model for P2P System without InfoRanking

In order to measure the performance improvement that InfoRanking leads to, we model the case in which no security scheme is used. In that case there is no information to help users to make a decision, so they just choose versions randomly.

Attackers pollute the system as much as they can, as there is no security scheme to limit their behavior. So each attacker inserts a corrupted copy to all clean versions and adds as many fake versions as possible. If each attacker is able to add k fake versions in the system the total number of fake versions in the system will be $N \cdot k$, where N is the number of the attackers. Moreover with the same assumption adopted in section 5.1 the total number of corrupted copies of a clean version is $N \cdot |V_c|$.

Users are assumed to make random selections uniformly across all the available versions, as there is no information to help them to decide which version to download.

In our model the worst case for benign users is considered, i.e., the attackers never leave the system, therefore the number of fake versions and copies inserted in the system will not decrease, in contrast to the clean copies that leave the system with rate ι. The vector $\hat{\pi}(t) = (m(t), x(t), z(t))^T$ is used to describe the system where $m(t)$ is the number of benign users searching for content, $x(t)$ is the the total number of clean copies of all clean versions, and $z(t)$ the total number of copies from fake versions shared by polluted benign users. Figure 8 shows the states of the system.

Since users who get corrupted copies of clean versions will delete them as soon as the download is finished, they can be regarded as being trapped in the state 'searching for content' until they download a clean copy. The fraction of this part of users is expected to be:

$$\frac{N \cdot c}{x(t) + z(t) + N \cdot (|V_c| + k)}$$

Similarly, the fraction of downloaders moving from 'searching' state to the state 'obtained a clean copy' or to the state 'obtained a copy of a fake version' are:

$$\frac{x(t)}{x(t) + z(t) + N \cdot (|V_c| + k)}$$

and

$$\frac{z(t) + N \cdot k}{x(t) + z(t) + N \cdot (|V_c| + k)}$$

Therefore, the evolution of $\hat{\pi}(t)$ can be described as:

$$\frac{d\hat{\pi}(t)}{dt} = \hat{\mathsf{Q}}(t) \cdot \hat{\pi}(t) + \begin{pmatrix} \xi \\ 0 \\ 0 \end{pmatrix} \tag{14}$$

where the transition matrix $\mathsf{Q}(t)$ satisfies:

$$\mathsf{Q}(t) = \begin{pmatrix} -\lambda \cdot A - \omega & 0 & \mu \\ \lambda \cdot B & -\iota & 0 \\ \lambda \cdot C & 0 & -\mu - \omega \end{pmatrix} \tag{15}$$

where

$$A = \frac{x(t) + z(t) + N \cdot k}{x(t) + z(t) + N \cdot (|V_c| + k)},$$

$$B = \frac{x(t)}{x(t) + z(t) + N \cdot (|V_c| + k)},$$

$$C = \frac{z(t) + N \cdot k}{x(t) + z(t) + N \cdot (|V_c| + k)}$$

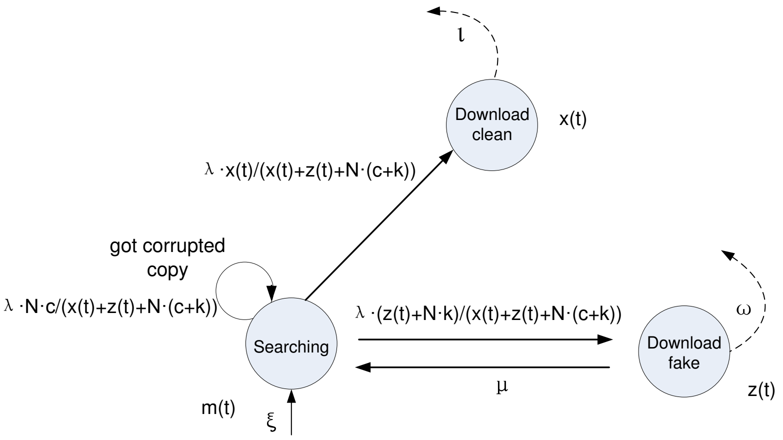
ι
Download clean
x(t)
λ ·x(t)/(x(t)+z(t)+N·(c+k))
got corrupted copy
λ ·N·c/(x(t)+z(t)+N·(c+k))
Searching
λ ·(z(t)+N·k)/(x(t)+z(t)+N·(c+k))
ω
Download fake
μ
z(t)
m(t)
ξ

Figure 8: Model describing system states without security scheme

In [25], it is shown that only when $\mu + \omega \geq \iota$, i.e. clean copies do not leave faster than corrupted copies, the probability that the system has clean copies is greater than 0. Here, we only consider the case $\mu + \omega \geq \iota$. By solving (14) and (15), the expected number of users in different states can be calculated as follows:

$$E(\hat{m}) = \frac{\xi + N \cdot \iota \cdot (|V_c| - \frac{k \cdot \mu}{\omega + \mu - \iota})}{\iota + \omega} E(\hat{x}) = \frac{1}{\iota} \cdot X E(\hat{z}) = \frac{k \cdot N \cdot \iota}{\mu + \omega - \iota} \tag{16}$$

where

$$X = (\xi + \frac{k \cdot N \cdot \iota \cdot \mu}{\mu + \omega - \iota} - \frac{\omega \cdot (\xi + N \cdot \iota \cdot (|V_c| + \frac{k \cdot \mu}{\mu + \omega - \iota}))}{\iota + \omega})$$

The probability that a new user selects a corrupted copy can be calculated as follows:

$$E(\hat{P}) = \frac{N \cdot \iota \cdot (\lambda + \omega) \cdot Z}{\lambda \cdot (\xi \cdot (\mu + \omega - \iota)) + Y} \tag{17}$$

where

$$Z = (|V_c| \cdot (\omega + \mu - \iota) + k \cdot (\mu + \omega))$$

$$Y = N \cdot \iota \cdot (k \cdot \mu + |V_c| \cdot (\omega + \mu - \iota))$$

While the mean time of a user in the system is:

$$E(\hat{T}) = \frac{\lambda + \iota}{\iota \cdot (\lambda + \omega)} + + \frac{N \cdot (\iota - \omega) \cdot (k \cdot (\lambda + \mu + \omega))}{\xi \cdot (\mu + \omega - \iota) \cdot (\lambda + \omega)} + + \frac{N \cdot |V_c| \cdot (\iota -}{\xi \cdot \lambda + \omega} \tag{18}$$

6.3 The Effect of InfoRanking

By comparing (11) and (17) it can be seen that the probability of a user that enters the system to get polluted is decreased by:

$$\frac{1}{\lambda} \cdot N \cdot \iota \cdot (\lambda + \omega) \cdot (A - \frac{1}{\xi + N \cdot \iota}) \tag{19}$$

where

$$A = (\frac{|V_c| \cdot (\mu + \omega - \iota) + k \cdot (\mu + \omega)}{\xi \cdot (\mu + \omega - \iota) + N \cdot \iota \cdot (k \cdot \mu + |V_c| \cdot (\mu + \omega - \iota)})$$

Figure 9 shows that difference.

It can be seen that when the number of attackers increases to a certain amount, the probability of getting polluted is extremely high if no security scheme is used. However, when InfoRanking is used, the probability of pollution increases almost linearly with the number of attackers, with a relativly flat slope. Thus, the probability of being polluted is still

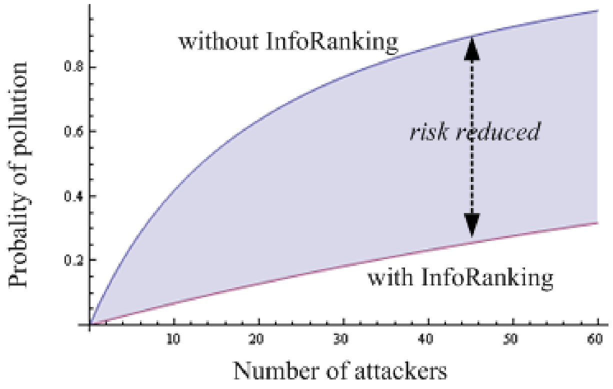

Figure 9: Probability of pollution

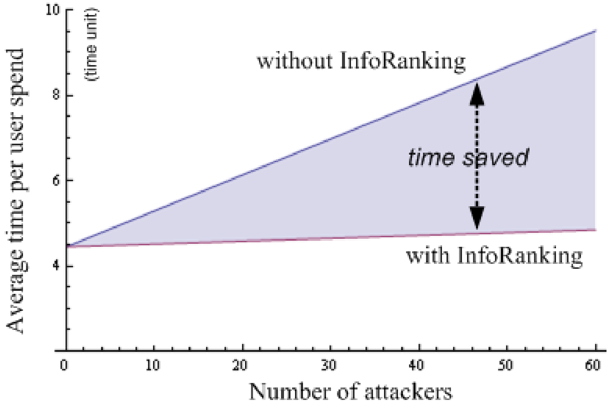

Figure 10: Average time spent by each user

under control when there are large numbers of attackers in the system. In this figure we have set $\iota = 0.3, \lambda = 0.5, \xi = 50, \omega = 0.1, \mu = 0.8, |V_c| = 1$ and $k = 5$.

Similarly, by comparing (13) and (18) we can see the gain in the average time each user spends in the system when InfoRanking is used:

$$\frac{N \cdot (\iota - \omega) \cdot B}{\xi \cdot (\omega + \mu - \iota) \cdot (\lambda + \omega)} \tag{20}$$

where

$$B = \lambda \cdot k + \iota - |V_c| \cdot \iota + (k + |V_c| - 1) \cdot (\mu + \omega)$$

The savings because of InfoRanking in the average time a user spends in the system is visualized in Figure 10, the shadow area shows the difference between a system without any security scheme and with InfoRanking. It can be seen that when there is no security scheme in the system, the average time spent by a user increases dramatically with the number of attackers, because he wastes a lot of time on fighting the pollution. However, when InfoRanking is used, the probability of pollution becomes low, and the average time spent in the system becomes stable.

7 Conclusion

In this paper we studied the effectiveness of InfoRanking in protecting P2P systems from content pollution. By ranking files rather than users and by considering positive votes only, InfoRanking not only can be easily implemented, but also does not require any incentive mechanism for engaging users in the voting procedure. Through analysis, fluid models and simulation we analyzed the performance of InfoRanking, considering various scenarios for the behavior of both benign users and independent attackers. In particular we considered the rate at which users enter and leave a system, the time they need to download a file as well as they time they need to detect pollution and we evaluated those factors in networks where attackers either insert corrupted copies of legitimate versions or share fake versions. Our findings show if users check for pollution relatively fast–less than $\simeq 2.4 *$ *time to download a file*–fake versions are completely eliminated from the system, therefore the only option for the attackers is to pollute legitimate version with fake copies. Nevertheless as fake files can be immediately detected this type of attack is only effective in network in which users enter with very small rate and leave fast.

Our future work includes the study of P2P file sharing systems under coordinated attack and the fine tuning of our model parameters to real conditions. Moreover we believe that with small modifications our model can be adapted to similar systems such as P2P live streaming, as well as to systems in which the popularity of a content item depends on users, e.g., application markets that rank applications based on user comments, auctions and user recommended bookmarks.

References

[1] Vanthournout K, Deconinck G, Belmans R. Building dependable peer-to-peer systems. *DSN 2004 Workshop on Architecting Dependable Systems*, 2004.

[2] Christin N, Weigend A, Chuang J. Content availability, pollution and poisoning in file sharing peer-to-peer networks. *Proceedings of the 6th ACM conference on Electronic commerce*, ACM, 2005; 68–77.

[3] Liang J, Kumar R, Xi Y, Ross K. Pollution in p2p file sharing systems. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2, IEEE, 2005; 1174–1185.

[4] Benevenuto F, Costa C, Vasconcelos M, Almeida V, Almeida J, Mowbray M. Impact of peer incentives on the dissemination of polluted content. *Proceedings of the 2006 ACM symposium on Applied computing*, ACM, 2006; 1875–1879.

[5] Resnick P, Zeckhauser R, Friedman E, Kuwabara K. Reputation. *Communications of the ACM* 2000; **43**(12):45.

[6] Josang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 2007; **43**(2):618–644.

[7] Cornelli F, Damiani E, di Vimercati S, Paraboschi S, Samarati P. Choosing reputable servents in a p2p network. *Proceedings of the 11th international conference on World Wide Web*, ACM, 2002; 376–386.

[8] Kamvar S, Schlosser M, Garcia-Molina H. The eigentrust algorithm for reputation management in p2p networks. *Proceedings of the 12th international conference on World Wide Web*, ACM, 2003; 640–651.

[9] Xiong L, Liu L. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *Knowledge and Data Engineering, IEEE Transactions on* 2004; **16**(7):843–857.

[10] Quercia D, Hailes S, Capra L. B-trust: Bayesian trust framework for pervasive computing. *Trust Management*, *Lecture Notes in Computer Science*, vol. 3986. Springer Berlin,Heidelberg, 2006; 298–312.

[11] Fotiou N, Marias G, Polyzos G. Information ranking in content-centric networks. *Future Network and Mobile Summit, 2010*, IEEE, 2010; 1–7.

[12] Cuevas R, Kryczka M, Cuevas A, Kaune S, Guerrero C, Rejaie R. Is content publishing in bittorrent altruistic or profit-driven? *Proceedings of the 6th International COnference*, Co-NEXT '10, ACM: New York, NY, USA, 2010; 11:1–11:12, doi:10.1145/1921168.1921183.

[13] Costa C, Soares V, Almeida J, Almeida V. Fighting pollution dissemination in peer-to-peer networks. *Proceedings of the 2007 ACM symposium on Applied computing*, ACM, 2007; 1586–1590.

[14] Walsh K, Sirer EG. Fighting peer-to-peer spam and decoys with object reputation. *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, P2PECON '05, ACM: New York, NY, USA, 2005; 138–143, doi: 10.1145/1080192.1080204.

[15] Costa C, Almeida J. Reputation systems for fighting pollution in peer-to-peer file sharing systems. *Peer-to-Peer Computing, 2007. P2P 2007. Seventh IEEE International Conference on*, IEEE, 2007; 53–60.

[16] Papaioannou T, Stamoulis G. An incentives' mechanism promoting truthful feedback in peer-to-peer systems. *Cluster Computing and the*

Grid, 2005. CCGrid 2005. IEEE International Symposium on, vol. 1, IEEE, 2005; 275–283.

[17] Kumar R, Yao DD, Bagchi A, Ross KW, Rubenstein D. Fluid modeling of pollution proliferation in p2p networks. *Proceedings of the joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '06/Performance '06, ACM: New York, NY, USA, 2006; 335–346, doi:10.1145/1140277.1140316.

[18] Lee U, Choi M, Cho J, Sanadidi M, Gerla M. Understanding pollution dynamics in p2p file sharing. *5th International Workshop on Peer-to-Peer Systems (IPTPS06), Santa Babara, CA, USA*, 2006.

[19] Yang S, Jin H, Li B, Liao X, Yao H, Tu X. The content pollution in peer-to-peer live streaming systems: Analysis and implications. *Parallel Processing, 2008. ICPP'08. 37th International Conference on*, IEEE, 2008; 652–659.

[20] Buragohain C, Agrawal D, Suri S. A game theoretic framework for incentives in p2p systems. *Peer-to-Peer Computing, 2003.(P2P 2003). Proceedings. Third International Conference on*, IEEE, 2003; 48–56.

[21] Mortazavi B, Kesidis G. Model and simulation study of a peer-to-peer game with a reputation-based incentive mechanism. *Proc. of ITA* 2006; **6**.

[22] De Paola A, Tamburo A. Reputation management in distributed systems. *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*, IEEE, 2008; 666–670.

[23] Varga A. *OMNeT++ Simulator Home Page.* http://www.omnetpp.org.

[24] Ramalhoto M, Amaral J, Cochito M. A survey of j. little's formula. *International Statistical Review/Revue Internationale de Statistique* 1983; :255–278.

[25] Zhang P, Helvik B. Towards green p2p: Understanding the energy consumption in p2p under content pollution. *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, IEEE Computer Society, 2010; 332–337.