Understanding Pollution Dynamics in P2P File Sharing[*]

Uichin Lee[†], Min Choi[‡], Junghoo Cho[†], M. Y. Sanadidi[†], Mario Gerla[†]
[†]*Department of Computer Science* [‡]*Department of Electrical Engineering and Computer Science*
University of California, Los Angeles *Korea Advanced Institute of Science and Technology*
[†]*{uclee,cho,medy,gerla}@cs.ucla.edu,* [‡]*min@kaist.ac.kr*

ABSTRACT

Pollution in P2P file sharing occurs when a large number of decoy files are injected into the P2P system. Since peers "serve" each other in the P2P file sharing system, it is obvious that pollution dynamics are closely related to user behavior. Therefore, we first conduct a human subject study to investigate user behavior. We identify the factors that are key to model user behavior, e.g., cooperativeness and awareness of pollution. Our results show that users are quite insensitive to pollution, and their behavior exhibits a bimodal distribution of the time interval between download and the quality checking of that download. We then propose a mathematical model to assess the impact of pollution on file popularity evolution. From our analysis we find that user "awareness" of pollution is a key factor in pollution dynamics. Finally, we study the impact of pollution on P2P traffic loads and show that in the worst case, pollution can quadruple the loads.

1. INTRODUCTION

Pollution has recently increased significantly in popular P2P systems such as KaZaA. A case that brought the problem to the fore occurred in 2003 when Madonna inserted warning messages into her new album and injected the polluted version into a P2P system. As a result, many of her fans were confronted with a foul-mouthed tirade. In fact, a number of companies, such as Overpeer[1] and Loudeye, specifically employ P2P pollution as a defensive technique to discourage illegal downloads [9]. By aggressively polluting the content and meta-data of genuine files and pouring as many polluted files as possible into P2P systems, they disguise false search results as genuine, significantly degrading the user experience and thus discouraging illegal downloads.

To analyze the approach above, we first examine how such a pollution attack works. A polluter may want to pollute a *topic* which is identified by a searchable string such as a song title. For example, assuming that we share the song "Hey Ya," users will search for it by querying "Hey Ya." They may then receive many results, i.e., multiple copies of the music with different encoding rates, types, etc. Here, "Hey Ya" is a topic and the different copies of the music files are distinct *versions*. For a given topic, a polluter creates polluted versions by using pollution techniques such as degrading quality or shuffling contents, and then injects such files into the system. When searching for some topic, a user will encounter the polluted files along with the genuine ones. Users cannot distinguish a genuine version from a polluted one before downloading it. After completely downloading the file, they may check whether the downloaded file indeed covers the topic of interest and whether the file is polluted.

Researchers have proposed a number of P2P user models to investigate the general pollution dynamics in P2P systems [1, 2]. However, a number of recent experimental studies show that the pollution level in the existing P2P network is significantly larger than what these models predict. For instance, reference [7] investigates the KaZaA network, one of the most popular P2P systems, and finds that more than 76.8% of 1,816,663 versions of the song "My Band" are polluted in the network, a level that far exceeds the prediction of these models under reasonable parameter settings.

The primary goal of this paper is to develop a simple yet reasonable extension to the existing P2P user models in order to better understand the pollution dynamics in P2P file sharing systems. Toward this goal, we first describe the results from our user survey that strongly indicate that even sophisticated P2P users often *unintentionally* help the polluter spread his polluted files because they are *unaware* that they have downloaded a polluted file. Based on this result, we then propose a new P2P user model that incorporates the pollution *awareness* of users (i.e., the fraction of users who notice the pollution in downloaded files and delete them). As we will see, our analysis shows that this awareness is one of the major factors in determining the final level of pollution; by incorporating this factor, the prediction of the model gets much closer to the observed level of pollution. To our knowledge, our work is the first study that considers user awareness and analyzes its impact on the overall pollution level in the network. Some of the key findings from our study are presented:

- We find that a significant fraction of users are rather *insensitive* to pollution. Even though a number of users check the quality of a file immediately after download (about 65% in our study), a large portion of users do not check quality for a long time after completion of download (often more than 12 hours). This results in a *bimodal* distribution in the interval.

- Furthermore, even after users check the quality of the downloaded file, a significant portion of them fail to notice that the file has been polluted. Our study shows that for certain types of pollution, more than 70% of the users fail to notice it, thus unintentionally spreading the polluted file to the network.

- Our analysis shows that the awareness of pollution is one of the major factors that affect the overall pollution level in the P2P network. For example, as user awareness decreases by a mere 20% (from 100% to 80%), the final pollution level can increase by a factor of 10 in certain cases.

[*]This work was supported in part by the National Science Foundation under Grant No. 0221528 and the Korean Science and Engineering Foundation under Grant No. M06-2003-000-10008-0.

[1]Overpeer was acquired by Loudeye in May, 2004.

- Our results also show the effect of pollution on the P2P network has the potential to *quadruple* the P2P traffic, because users often try to re-download a genuine copy of the polluted file that was just downloaded.

The reminder of this paper is organized as follows: Section 2 summarizes related work; Section 3 presents results of a human subject study; Section 4 describes our analytic pollution model and presents its results; Section 5 discusses the impact of pollution on P2P traffic load; and finally, we conclude the paper and discuss future work.

2. RELATED WORK

Good et al. [3] studied usability and privacy issues of KaZaA; users who accidentally or unknowingly allow their private files to be shared, potentially disclose their private information. The authors found that a large number of users were unable to determine what they were sharing. Thus, it is possible that users unknowingly help spread polluted files if they are unaware of sharing polluted files. In this paper, a human subject study was performed to confirm such observation. From this, we found that users are indeed not error-free in recognizing pollution.

Christin et al. [1] addressed content availability taking into account pollution impact. The authors described possible strategies of pollution as a random decoy attack or a replicated decoy attack. While a random decoy attack employs a massive number of decoys, a replicated decoy attack injects numerous replicas of the same decoy. Given that typical P2P networks limit the number of returns that a given query can yield, the authors showed that replicated decoy attacks are more efficient than random decoy attacks. Here, the authors assumed polluted copies do not propagate, and random decoy injection does not change the availability of usable files. The authors noted that a combination of random and replicated decoy attacks would be difficult to detect and would significantly decrease the content availability of the file. In this paper, not only do we show that polluted files or decoys do propagate, but we also find that this makes the attack subsequently lessen the availability of usable files.

Later, Dumitriu et al. [2] made the first attempt to model the dynamics of P2P file pollution attacks. The authors assumed that a polluted node removes a polluted file within a certain amount of time; i.e. not only does a user always detect the polluted file, but he also deletes it. This assumption implies that in the end only the polluters have the polluted files and thus polluted copies cannot spread over the network. Our paper, unlike previous work, shows that polluted files indeed spread from user to user over the network – mainly due to the lack of user awareness. In addition, we show that pollution has a significant impact on P2P traffic loads.

3. USER BEHAVIOR STUDY

In this section we report on a human subject study in which we tried to understand the general user behavior in a P2P network. This study was conducted in two stages. In the first stage, we surveyed a total of 30 students at UCLA and KAIST[2] to get a sense of their familiarity with the P2P network and their general usage patterns. In the second stage, we asked the 30 participants to use a modified version of a popular P2P client, so that we could observe their usage behavior in more realistic settings. We now explain the settings and our findings from these studies in detail.

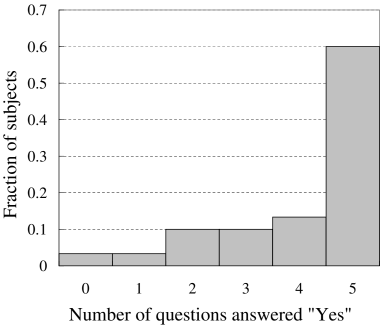


Figure 1: P2P familiarity index

3.1 User Survey

Our survey questionnaire consisted of two main parts. In the first part we tried to evaluate the familiarity of our participants with the P2P systems because our findings could be significantly biased by the familiarity of our participants with the systems. In the second part we wanted to perceive a general sense of how the participants used and how they handled downloaded files. In this way, we could identify the key factors that affect pollution dynamics in the P2P network.

3.1.1 P2P Familiarity

More precisely, in the first part, we asked the participants the following five questions: (Q1) Have you ever used P2P file sharing? (Q2) Do you frequently share files with P2P systems? (Q3) Do you know how to enable or disable sharing local files? (Q4) Do you know how popular P2P software works? (Q5) Do you know about multi-part downloading or swarming? These questions were designed such that a user with a more detailed knowledge of the P2P system would answer “Yes” to higher number of questions. The results of these questions are shown in Figure 1. From this figure we can see that the majority of our participants, i.e., 60%, said “Yes” to all five questions. This result is not surprising because most of our participants are graduate students in the Computer Science Department. Later, we will discuss the implication of this bias in our user group.

3.1.2 P2P Usage Pattern

In the second part of our user survey, we tried to understand how the users download files in a P2P network and how they handle the downloaded files. In general, P2P client usage can be broken down into three stages: download preparation, download, and post-download stages. In the preparation stage, a user sends a query and selects a file to download. In the downloading stage, the user checks the status of the download and sometimes goes back to the first stage if the download speed is too slow. Lastly, in the post-download stage, the downloaded file is checked and the user makes a decision to share the file or not to share the file. In our survey, we asked a few questions related to each of these three stages.

For the preparation stage, we asked our participants to determine the most important criteria used making a download decision. For this question, the vast majority of our participants, 57%, indicated that the quality of the file was the primary criterion. The availability of a file was ranked a

[2] Korea Advanced Institute of Science and Technology

distant second at 20%,[3] with similar number of participants (slightly less than 20%) indicating file size[4] as their primary criterion.

For the download stage, we asked participants three questions: (1) How often do they check the status of a download? (2) Do they cancel downloads due to slow speed? (3) Do they usually download multiple files simultaneously (either on the same topic or on different topics)?. For the first question, 41% answered that they frequently check the status during a download, while 24% of the users said that they just leave the download alone and only check the status some time later when the download has likely finished. The remaining 35% answered that it depends on the file size. If the size is small, they may check the status frequently, but if not, they may check after a while. For question 2 and 3, 83% answered that they do start a new download process when the speed is too slow. 63% also indicated that they often download multiple files simultaneously.

For the post-download stage, we asked the following three questions. First we asked if participants are usually “cooperative” in sharing downloaded files, for which 43.3% said “Yes.” Second, we asked if they had ever downloaded polluted files before, for which 70% answered “Yes.” Interestingly, many of the users (about 30%) reported that they actually had an experience in which they initially thought they had downloaded a genuine version and decided to keep it, but later realized that the file was, in fact, polluted. This was a surprisingly large number given the technical sophistication of our participants. Even with their deep understanding of the P2P system and their full awareness of the pollution problem, our participants sometimes failed to recognize polluted files. Finally, we asked if they normally re-downloaded files when they recognize a polluted file. 23% indicated that they usually re-downloaded files if they recognized pollution and 57% said that it depended on the size of the file.

In summary, from our user survey, we found that: (1) even sophisticated P2P users sometimes fail to recognize polluted files, (2) many users do not check the quality and authenticity of a downloaded file immediately after the completion of download, (3) not all users are cooperative in sharing downloaded files, and (4) users make their download decisions primarily based on the expected quality of a file.

3.2 Experimental Measurement

The most surprising result from our user survey was that even technically sophisticated users sometimes fail to recognize the pollution in their downloaded files. We also found that quite a large number of users do not check the quality of their downloaded file even long after the completion of its download. We wanted to investigate these issues further in more realistic settings, so we conducted the following measurement study.

In a measurement study of the survey, users were asked to use a modified P2P client which connects to a server and allows them to download files from the server. We performed this test for a period of one month in October, 2005. Users were given a list of files to download; we instructed them to check their downloaded files and answer whether files were polluted or not. To make our setting close to actual P2P systems, downloading speed was randomly chosen to fall between 50K and 1Mbps. Overall it took a user less than ten minutes to download a file. Using this setting, we were mainly interested in measuring the following two parameters:

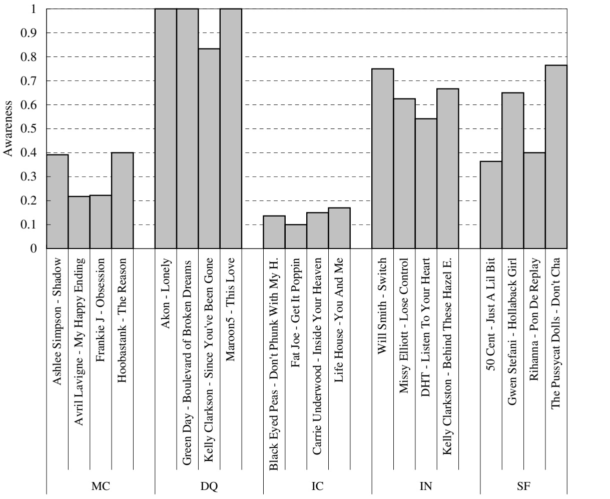


Figure 2: Awareness of pollution with different types of pollution techniques

- *Awareness probability*: the fraction of users who recognize pollution in a downloaded file
- *Slackness distribution*: distribution of intervals between download completion time and quality checking time.

For this measurement, we chose 20 currently popular songs and created polluted versions by tampering with either their meta-data or with their content according to [7]. Meta-data was falsified by changing the file name or modifying the description of the file content, e.g., bit rate, and we call this modification *MC*. To pollute a file content we degraded the content quality (DQ), made the files incomplete (IC), inserted noise (IN), or shuffled the content (SF). After seeding the server with both genuine and polluted files, we asked users to randomly select files from the server, download the files, and judge whether the downloaded files were polluted or not. We also asked our users to indicate their familiarity with each downloaded topic to access the impact of their familiarity on the awareness of pollution.

Figure 2 shows the results of this user awareness measurement for each pollution type. With meta-data modification (MC) pollution, the users showed less than a 50% awareness, which is mainly due to the lack of familiarity with the selected songs. It is interesting to note that there were quite a few users who answered *not polluted* even though they indicated familiarity with the songs. As expected, users easily detected degraded quality, i.e., DQ. On the other hand, in the case of incomplete files (IC), users exhibited very a low awareness *regardless of familiarity*. The songs used for IC were generated by cutting off 30 to 60 seconds of the songs from the beginning or at the end and also by applying a fade-in/out filter. Even though more than 30% of a song is cut off, many participants who indicated familiarity with the songs failed to recognize this. In part, this explains the high level of pollution observed in KaZaA [7] where the authors assumed that a file is polluted if its length is not within +10% or -10% of the official CD version. For inserted noise (IN) pollution, more than 60% of users recognized their version as polluted. Because noise was inserted every 20 seconds, we conjecture that 40% of users listened to the music less

[3] Availability means the number of users who currently have the file.

[4] Our participants preferred files that were smaller in size.

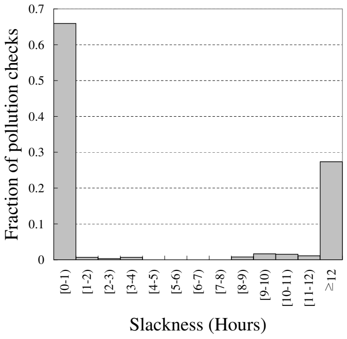


Figure 3: Distribution of slackness

carefully and then made their decisions. Finally, in the case of shuffled content (SF) pollution, quite a few users failed to realize pollution regardless of familiarity. Interestingly, if songs have a fast beat, e.g., hip-hop or rap, users showed lower awareness. Note that we also plotted the same graph using only participants who claimed familiarity with the files they checked, and the results have the same tendencies as shown in Figure 2.

We measured "slackness" in checking the quality or the authenticity of a downloaded version by recording the elapsed time between download completion and pollution checking. The total number of pollution checks was 981 out of 1200 expected checks, or 82%. Not all subjects downloaded all the test files due to program errors or their own negligence. Figure 3 shows the histogram of slackness. It is interesting to note that 65% of the checking intervals were within an hour, or [0-1), and 27% were longer than 12 hours. This implies that users either wait for download completion and then check, or leave the download alone and check it some time later. Note that this result may be biased because we reminded users every day. Thus, we suspect that the fraction of pollution checks that would normally happen during [0-1) hour would be lower than the value observed in Figure 3.

From the user behavior tests we concluded the following: (1) P2P users are lacking in pollution awareness; (2) slackness distribution shows a bimodal form.

4. POLLUTION MODEL

In this section we develop an analytic model to study pollution dynamics by extending [2] and incorporating what we learned from the survey and experiment reported in the previous section. We assume that there are M users. Every user maintains only one version of a topic.[5] Initially, there are G_0 users with genuine copies and B_0 users with polluted or $bogus$ copies. The users with these initial copies never leave the P2P network. Other users without an initial copy download the files over time through the following process:

1. At each time step k, a user who never downloaded a version before becomes interested in the topic, issues a query and downloads a version with probability s_k, a measure of the "interest level" for the topic.
2. Once the file is downloaded, the user checks its authenticity after an interval t. We assume that the interval t is a random variable with an upper bound L called the maximum slackness. We refer to this as the "slackness" distribution. Until the user checks the validity of a file, the downloaded file is shared in the P2P network. [6]
3. After checking the validity of the downloaded version, if the user realizes that the version is bogus, the user deletes it. The user, however, is not error-free in detecting the authenticity of the version. Even if a file is bogus, the user may not notice the pollution and may believe that the version is authentic with probability $1 - p_a$. Thus p_a is a measure of the user's "awareness" of pollution.[7] If the user does notice the pollution and delete the file, in the next time step, he tries to re-download a file with probability p_r, and repeats the process in step 2.
4. After checking the validity of the file, if the user believes that the file is authentic (either because the file is indeed authentic or because he has failed to detect the pollution), the user makes a decision on whether he will continue to share the file or not. With probability p_c, a measure of "cooperativeness," the user continues to share the file. With the remaining probability $1 - p_c$, the user leaves the P2P network.

At time step k, let G_k and B_k denote the number of users who currently hold genuine and polluted (*bogus*) copies respectively. Let D_k denote the total number of users who have downloaded a file by step $k - 1$. Since $M - D_k$ users have not ever tried, then at time step k, a fraction s_k of $M - D_k$ users will download a file. Therefore the sequence D_k satisfies the following relationship.

$$D_{k+1} = D_k + (M - D_k)s_k \tag{1}$$

At time step k, let g_k and b_k denote the total number of users who download genuine and polluted versions respectively. At time step k, a total of $(M - D_k)s_k + r_k$ users will download a file, including both brand-new trials, $(M - D_k)s_k$, and retrials due to pollution, r_k. Assuming that the polluter can pollute the meta-data of the file such that users randomly select a source, the probability of selecting a genuine file is given as $p^G_k = G_k/(G_k + B_k)$.[8] Thus we have

$$g_k = ((M - D_k)s_k + r_k)p^G_k \tag{2}$$

$$b_k = ((M - D_k)s_k + r_k)(1 - p^G_k) \tag{3}$$

Let t be a random variable such that a user checks the downloaded file after t slots and p^S_t denote its *slack* probability. Thus, uncooperative users leave the system after j slots with probability p^S_j, and the total number of genuine files at time step $k + 1$ can be written as follows

$$G_{k+1} = G_k + g_k - (1 - p_c)\sum_{j=1}^{L} g_{k+1-j}p^S_j \tag{4}$$

Suppose a user downloads a polluted file. If the user becomes aware of the pollution, then he will delete the file.

[5] We assume that the polluter has the same capacity as other peers. The extended version of this paper [6] describes the service capacity model using a branching process with immigration which considers polluters with higher capacity.

[6] Most P2P clients, e.g. BitTorrent and eDonkey, support multi-part downloading or swarming and thus files or part of files are shared by default.

[7] This probability can be written as $\mathbb{P}$[*a user recognizes as polluted* | *a file is polluted*]=p_a. We assume that $\mathbb{P}$[*a user recognizes as genuine* | *a file is genuine*]=1.

[8] Quality can only be inferred through meta-data. If the polluter fakes such information, as we learned from a user behavior study, a user selects a file based on its availability.

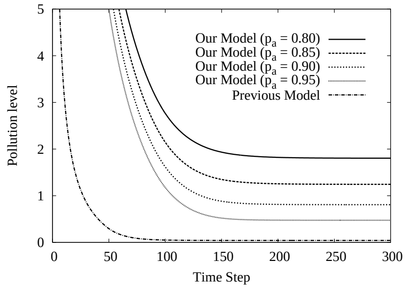


Figure 4: Pollution level as a function of time

On the other hand, if he fails to recognize the pollution, he will share the file with probability p_c. Because these events are independent, he deletes the file with probability $p_D = p_a + (1 - p_a)(1 - p_c)$. Such deletion happens at a time slot j with probability p_j^S. Therefore we have

$$B_{k+1} = B_k + b_k - p_D \sum_{j=1}^{L} b_{k+1-j} p_j^S \tag{5}$$

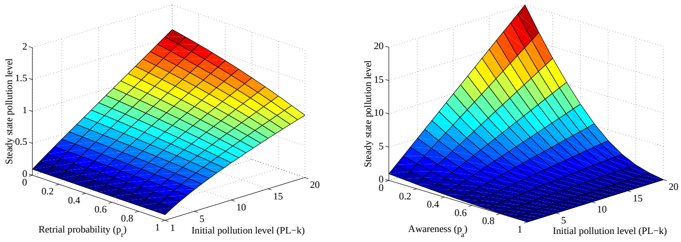
Finally, retrials only happen when users are aware of pollution (p_a) and also want to download files again at the next time step (p_r).[9] Thus the number of retrials at $k + 1$ time step is

$$r_{k+1} = p_a p_r \sum_{j=1}^{L} b_{k+1-j} p_j^S \tag{6}$$

Analytic Results

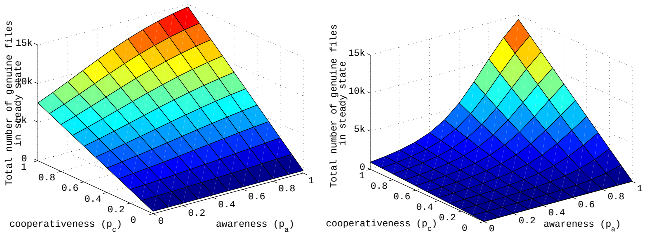
Let the total number of users $M = 15,000$. We use the measured slackness distribution from our human subject study with an upper bound $L = 48$. The interest factor s_k was set to $1/24$ such that each peer is interested in downloading a file on an average of once per 24 hours. For ease of illustration, we assume that those who download a polluted file always try again, or $p_r = 1$, which allows us to observe the worst case from the polluter perspective. In addition, we assume that a random user cooperates with probability $p_c = 0.25$. For awareness we use the measured value for the song "The Pussycat Dolls-Don't Cha," or $p_a = 0.76$. Unless otherwise mentioned we use the above as a default setting. We derive our results by iteratively solving the equations in the preceding section. To measure the efficacy of pollution, we define a *pollution level* as the ratio of the number of polluted copies to the number of genuine copies for a given time slot.[10] The "initial" pollution level is denoted as PL-k where k is the ratio. Note that the final pollution level is also referred to as the "steady state" pollution level.

Let us first compare our model with the previous model [2] where users are perfect in recognizing pollution, but slack in deleting polluted files. To this end, we use the initial pollution level PL-20, and use different values of awareness from 0.80 to 0.95 for our model. Figure 4 shows the pollution level as a function of time. Note that the upper bound of the pollution level is 20 because the polluter starts with PL-20. Since users were accurate in recognizing pollution in the previous model, the pollution level reaches almost zero as times goes on. On the other hand, our model shows that polluted files indeed spread due to the lack of awareness; e.g., while the final pollution level of the previous model is 0.05, our model shows that the final pollution levels are 1.2 and 1.8 for $p_a = 0.85$ and $p_a = 0.80$, respectively. In addition, from the graph we can see that as awareness decreases, the pollution level increases. Thus, such a high level of pollution in KaZaA [7] can be explained using our model.



(a) Retrial with $p_a = 0.76$ (b) Awareness with $p_r = 1$

Figure 5: Steady state pollution level



(a) Pollution level PL-1 (b) Pollution level PL-20

Figure 6: Number of genuine files in steady state as a function of cooperativeness and awareness

We then study the effectiveness of *increasing* the initial pollution level by the polluter. To understand this we consider both retry probability (p_r) and awareness (p_a) with two different pollution levels: PL-1 and PL-20. Let us first examine the retry probability. Figure 5(a) shows that as retrial probability increases, increasing k shows much less than linear improvement. Thus the more that users are impatient, i.e., exhibiting low p_r, the more the polluter is successful in polluting files. The results for awareness are shown in Figure 5(b). As awareness increases, a higher k does not provide the polluter much improvement. If the polluter's goal is to achieve a certain level of pollution in steady state, then without lowering user awareness he can hardly achieve such a goal. Put differently, given that the polluter has a limited number of machines which in turn bounds his initial level of pollution, only by lowering user awareness he can perform a large-scale attack.[11] For example, with PL-20 lowering awareness 20% (from 100% to 80%), we can increase the final pollution level by a factor of 10.

Finally, we investigate the relationship between cooperativeness and awareness in steady state. We plot the results of PL-1 and PL-20 in Figure 6 with different values of p_a and p_c. Interestingly, when the level of pollution is low (PL-1), both p_c and p_a are almost linearly proportional to the number

[9] For ease of formulation, we assume that a user is memoryless, and thus his behavior follows a geometric distribution with success probability $1 - p_r$.

[10] Note that G_k and B_k increase proportional to the number of files and thus changing absolute numbers while preserving ratio does not influence the results assuming that the numbers (G_k and B_k) are much smaller than the total number of users.

[11] The polluter can only control awareness unlike other parameters such as retry probability and cooperativeness.

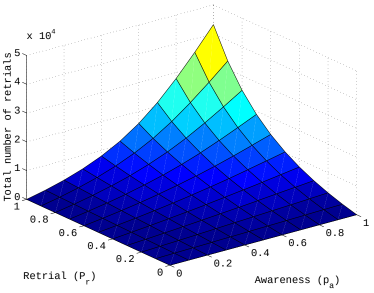


Figure 7: Total number of retrials as a function of retrial probability p_r and awareness p_a

of genuine copies, but when the level of pollution is high (PL-20), given that we have fixed p_c, as p_a increases, the number of genuine copies grows much faster. To understand why this happens, we need to take a look at Eq. 4 and Eq. 5. It is obvious that manipulating p_c influences both G_k and B_k, and thus G_k is a linear function of p_c. However, increasing p_a adversely affects the number B_k, which in turn makes users try again to download a genuine version. As awareness of pollution p_a increases, the number of retrials r_k also increases. Since the increment rate of r_k is directly related to the level of pollution, the impact of r_k in PL-1 is relatively small compared to that of r_k in PL-20. Thus we conclude that as the level of pollution increases, awareness becomes much more important than user cooperativeness for the growth of genuine copies.[12]

5. IMPACT ON INTERNET TRAFFIC LOAD

As soon as a user recognizes that he has downloaded a polluted version, he is likely to repeat the download, thus causing additional network traffic. How much traffic can pollution generate? To make such an assessment we first need to consider topic popularity which reflects the interest rates of users. According to a measurement study of KaZaA [4], only a small percentage of total topics are queried frequently. Further, the study revealed that the popularity of KaZaA files has short lifetime and ironically those popular files are the targets of the polluters. A pollution attack happens repeatedly and therefore, this will result in a large number of unnecessary downloads. The number of unnecessary downloads at time step k can be determined using Eq. 6. Therefore, until we reach steady state, say at time step t_s, the total number of retrials is

$$\sum_{k=1}^{t_s} p_a p_r \sum_{j=1}^{L} b_{k+1-j} p_j^S \tag{7}$$

To study how severe an impact a pollution attack would have on the number of retrials, we plot Eq. 7 as a function of awareness p_a and retrial probability p_r with PL-15 in Figure 7. To our surprise, in the worst case the number of retrials has more than *triple* the number of trials and this could thus *quadruple* the P2P traffic load. Considering the fact that 60% of the traffic on the Internet is made up of P2P activity, a pollution attack is likely to have a significant impact on Internet traffic load.[13]

6. CONCLUSION

In this paper we studied detailed P2P user behavior through a human subject study. We showed that users indeed exhibited *low awareness* with most types of pollution. In addition, they checked their download either immediately upon its completion, or a long time later, and thus slackness has a *bimodal distribution*. Guided by the user behavior study, we developed a mathematical pollution model to better understand pollution dynamics. From the analysis we showed that *awareness* is a key factor in pollution dynamics; thus, a polluter must lower user awareness to perform an effective large-scale attack. Finally, we discussed the impact of pollution on the network traffic loads. We showed that attacks on popular files could *quadruple* the P2P traffic loads.

There are several interesting avenues for future work on this subject. First, we are interested in monitoring user behavior when downloading files other than music, e.g. movies and software. We suspect that user behavior will be different because the file sizes for such topics are typically much larger than music files. Second, we could conduct further research on other parameters used in our model, i.e., cooperativeness and retrial probabilities, which will bring us more insight into pollution dynamics. Finally, it will be also interesting to design a reputation system reflecting the observations in this paper. For instance, most proposed reputation systems [5, 8] assumed that "honest" users are without error-free in recognizing quality of files but as we have shown, that is not the case.

REFERENCES

[1] N. Christin, A. S. Weigend and J. Chuang, Content Availability, Pollution and Poisoning in Peer-to-Peer File Sharing Networks, *ACM E-Commerce Conference (EC'05)*, June 2005.

[2] D. Dumitriu, E. Knightly, I. Stoica, and W. Zwaenepoel, Denial-of-Service Resilience in Peer-to-Peer File Sharing Systems, *In Proc. of ACM SIGMETRICS'05*, June 2005.

[3] N. Good and A. Krekelberg, Usability and Privacy: A Study of KaZaA P2P File-Sharing, *In Proc. of SIGCHI'03*, 2003.

[4] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, J. Zahorjan, Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload, *In Proc. of SOSP'03*, October 2003.

[5] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina The EigenTrust Algorithm for Reputation Management in P2PNetworks, *In Proc. of WWW'03*, May 2003.

[6] U. Lee, M. Choi, M. Y. Sanadidi and M. Gerla, Understanding Pollution Dynamics in P2P File Sharing, *UCLA CSD Techical Report*, October 2005.

[7] J. Liang, R. Kumar, Y. Xi and K. Ross, Pollution in P2P File Sharing Systems, *In Proc. of INFOCOM'05*, May 2005.

[8] K. Walsh, E. G. Sirer, Fighting Peer-to-Peer SPAM and Decoys with Object Reputation, *In Proc. of P2PECON'05*, August 2005.

[9] Loudeye Pushes P2P Antipiracy Tech `http://www.technewsworld.com/story /34063.html`

[12] Note that this is only true with a certain level of cooperation. If users are not cooperative at all, genuine copies cannot spread.

[13] CacheLogic reported the statistics by measuring the traffic on the Internet by the end of 2004.