ISSN 2250-3153

An Robust Pollution Control Strategy for P2P File Sharing System

G. N. Purohit, Urmila

Reasearch Scholar
Department of Computer Science, Banasthali University, Bansthali, 304022, India
gn_purohitjaipur@yahoo.co.in, Urmil.malik84@gmail.com

Abstract- Despite being currently one of the main internet application.P2Pfile sharing has been hampered by content pollution attack. To tackle this problem, we introduce a novel pollution control strategy that consisting in adjusting the rate in which content disseminated, according to content version reputation. The proposed strategy is modeled and evaluated using simplifying assumptions. Then inspired by classic distributed design, we propose a pollution control mechanism that implement such a strategy .The mechanism is evaluated in terms of the delay imposed on non polluted version dissemination when the version is polluted and the negative impact that collusion attack can impose on the reputation system upon which our mechanism is built.

I. INTRODUCTION

Peer to Peer (p2p) file sharing system has been widely and largely adopted for content dissemination among users in the internet. Recent study shows that p2p file sharing is responsible for large amount of traffic on internet. Despite their popularity p2p file sharing application have been suffering from different kind of attacks i.e. pollution, poisoning attack. In the p2p network content are made available as titles. Version of these titles can be uniquely identified by applying a hash function over the metadata. Content pollution consisting of distributing contents that range from mislabeled or accidentally damaged files to malicious decoys(virus and other malware).User can unable to distinguish incorrect/polluted from the correct one may download multiple version before giving up or obtaining the correct desired content. Pollution spread the malicious decoysand assumes that user will attempt to obtain a title through repeated downloads which result into unnecessarily consumption of client and network resources. Previous study and evidence shows that reputation systems are robust solution to protect p2p network from malicious attack. In a reputation system the quality of the resource /peer is determined by a user based on historical information from other users. In the bittorrent to the best of our knowledge only a set of adhoc solutions designed specifically to mitigate the pollution in user communities. These are simplistic solution such as discussion forum where user can post testimonials about content, reporting mechanism to notify the community administrator and voting mechanism to automatically isolate suspicious content .This approach requires a non negligible moderation effort through manual inspection of the content. Beside there are no mechanism to provide incentive for users to cooperate against pollution. Considering the limitation of the above solutions this paper proposes a conservative strategy and a mechanism to control content pollution dissemination in bittorrent. This approach counts the positive and negative vote assigned to the content to classify them as polluted or non polluted content. In this paper we propose and analyze novel pollution control strategies that handle early stage of pollution dissemination. This pollution control strategy relies on the notion of rate limiting the dissemination of content according to the reputation of content.

II. RELATED WORK

Content pollution is a subject that has recently received a great attention of the research community. Eigentrust[12] is an algorithm used to establish a global trust value for each peer which could then be used for selecting reliable sources. One practical limitation is the need of a set of pretrusted peers to operate adequately. Liang et.al.[5]propose the creation of a blacklist based on evaluation of title meta data collected in a p2p network. The list is composed of IP range labeled as polluters and built according to the density of corrupted files made available. The success of the approach depends on the frequent download of metadata, potentially leading to substantially overhead. In a continuation of the previous investigation [3], a blacklist is created by a global system to keep track of sub network reputation. In both investigation there are limitations in characterizing a sub network as polluter based on the high density of polluted versions, many participants of p2p network are connected to the internet via Network Address Translation in Internet Service Provider, besides to forge IP addresses is a well known attack technique, which could be used to trick the mechanism. A subsequent work proposed the Credence reputation model [9] the evaluation of the model using the Gnutella protocol led to satisfactory result. Credence is based on the reputation score that are assigned to content in order to govern the dissemination in the system. However this strategy does not prevent newly published, polluted content to the widely disseminated when there is little knowledge available to form a opinion about their reputation .This is worsened when new polluted contents are quickly publishes. Scrubber [11] is a reputation system similar to Credence. It adopts in comparison a more aggressive approach to penalize content polluters .This allows a faster convergence to a small percentage of daily downloads of polluted content. The same author later discuss the drawbacks of Scrubber and propose Hybrid [4].The latter combines object reputation with peer reputation to improve

pollution identification and assists user decision about pollution. Hybrid addresses the problem (not solved by Scrubber) of cleaning (that is detecting and restricting the dissemination of) polluted objects shared by peers that only occasionally upload them and thus manage to keep good reputations. However, despite the progresses on pollution control attained by the aforementioned proposals, they fail to limit the spread of polluted content in its early stage of dissemination.

FUNNEL: The Conservative Model: The mechanism FUNNEL operates by controlling the distribution of content according to the votes. It calculates the reputation of a content version and adjust the number of downloads according to this.
FUNNEL uses a binary voting mechanism to calculate the reputation of a content version i.e. E[W]. According to the FUNNEL when a user finishes a downloading then the user issues a positive /negative feedback about the content and adjusts the number of concurrent downloads according to the reputation scale.
Subjective Logic[13] is used to calculate the reputation score and consequently assume values in the interval[0,1].To calculate the reputation score of a content the mode consider positive and negative votes assigned by the user.
$E[\omega] = r+2a/r+s+2$

Funnel keeps track of the current number of downloads and employs $E[\omega]$ to determine if a download request should be authorized. In this context, the mechanism introduces two key variables: A and D, defined as follows. A represents the maximum number of concurrent downloads to be allowed, whereas D represents the current number of downloads taking place. The value of A is calculated according to Equation 1, in function of the corresponding version reputation ($E[\omega]$). The value of D is incremented when a download begins, and decremented when it ends. Finally, A and D are employed by the pollution control mechanism to allow new download requests to proceed if $D < A$.

$$A = E[\omega] \times (A\text{free} - A\text{min}) + A\text{min} \quad (1)$$

Amin represents the minimum number of concurrent downloads to be allowed (which is relevant when $E[\omega] = 0$). A free, in turn, corresponds to the number of concurrent downloads to be allowed when $E[\omega] = 1$. If so, there are only positive votes, and the content is assumed to be non-polluted. Consequently, the restriction on the number of concurrent downloads is lifted. Nevertheless, by analyzing Equation 1, we find that $E[\omega] = 1 \leftrightarrow 2a = n+2$. The only solution for this equation, given that $a \in [0; 1]$ and $n \in N$, is $a = 1$ and $n = 0$ (absence of negative votes). To address this issue, we introduce a threshold σ, with $\sigma \in [0; 1]$, to "free" downloads even in the presence of negative votes. When $E[\omega] = \sigma$, the content is assumed to be non-polluted and infinite concurrent downloads may take place.

When $E[\omega] \geq \sigma = 0.9$, an arbitrary number of downloads can take place; if new negative votes are provided, the reputation score could fall below 0.9, with the limitation being reinstated .If a new download is possible, the values of A and D are compared (and also of $E[\omega]$ and σ). If $D < A$ or $E[\omega] \geq \sigma$, new downloads can be allowed.

In order to employ the model on BitTorrent communities, some aspects need to be addressed: how to (*i*) estimate the number of concurrent downloads taking place (*D*), (*ii*) deny a download request, (*iii*) avoid duplicate votes, and (*iv*)encourage users to vote. In this work, we assume that most users will vote correctly A BitTorrent tracker is modified to implement Funnel, the model is instantiated aiming at "closed" BitTorrent communities (*i.e.*, require registration). A tracker in BitTorrent enables peers (users) to contact each other and switch pieces of files. Peers frequently ask for a list of other peers to connect to. Since users need to register, Funnel assumes Sybil attacks [3] are not feasible. This assumption guarantees that a unique user does not have enough autonomy to manipulate the reputation of a content version. Furthermore, the tracker can use the user's registry to identify later interactions..

A. Estimating the number of concurrent downloads
The strategy adopted by the tracker to estimate the value of D considers the number of peers downloading a specific torrent and is based on the frequent interaction of peers. Every time a peer P asks for a list of peers (PeerList), the tracker associates a temporary registry with P; if the registry expires, it is removed by the tracker. The value of D is assumed to be the number of valid registries. This strategy is susceptible to collusion attacks, where malicious peers try to manipulate the estimated D value. Since the value of D can be overestimated by the attackers, eventually the condition $D < A$ is false and the mechanism denies as subsequent download requests, causing a denial of service. However, this seems to be interesting only in the scenario where malicious peers try to harm the dissemination of an authentic content. To address this problem, when $D \geq A$, the tracker allows new download requests to proceed with a specific probability.

B. Denying a download request
To deny a download request, the tracker manipulates the PeerList returned to peers. For example, if the tracker understands that a download request must be denied ($D \geq A$), it returns an empty PeerList to the requester peer. However, the BitTorrent Distributed Hash Table (DHT) extension might interfere with this strategy. It enables peers to bypass the tracker and disseminate PeerLists using DHTs. We quantified the proportion of published torrents over four communities and found out that this value was no larger than 9.12%. To reduce the effect of DHTs on our mechanism, the tracker may prohibit the publication of torrents supporting this extension.

C. Avoiding duplicate votes
Considering the assumption that the tracker maintains registries for users and distinguishes its interaction with each user, it employs a table indexed by the user's identity to determine if a specific user has already voted in a torrent.

D. Encouraging users to vote
There are two options to encourage users to vote: reward those who vote or punish the ones who do not. In order to avoid votes without real content inspection, rewards are not given to the users who vote. Instead, the tracker adjusts the PeerList size according to the proportion of torrents a peer joined and provided a vote. Let N be the original PeerList size, V the number of

torrents a peer has cast a vote, and R the number of joined torrents, the new PeerList size is adjusted by the tracker to $N \times \min\{V+1\ R\ ,\ 1\}$. Given that a vote is accepted only if the peer joins the torrent, $V \leq R$.

III. EVALUATION

This section presents an experimental evaluation of FUNNEL. The purpose of this evaluation is to provide answers to three fundamental questions: (i) the negative impact caused by the mechanism in a scenario where a non-polluted content is disseminated or, in other words, the overhead imposed by FUNNEL with non-polluted contents; (ii) the effectiveness of FUNNEL in slowing down the dissemination of polluted contents to normal users; (iii) the negative impact of collusion attacks performed by malicious peers against the mechanism. Each of the scenarios employed in the experimental evaluation comprised the dissemination of a single content version (one torrent), due to two reasons. First, unlike other approaches, FUNNEL spawns individually and independently to every content version. Second, although FUNNEL may influence the user's choice of torrent – assuming there is more than one torrent for the same content – by decreasing the number of leechers (and indirectly, of seeders), the choice will be affected by multiple subjective factors. These include information contained in the meta-data (e.g., publisher alias), number of seeders and torrent age. We do not know of any study that analyzes the rationale behind this choice and safely establishes a user behavior pattern.
Let I, C and M denote, respectively, the number of initial seeders (either honest or malicious) that start a swarm and remain available during the entire experiment, and the total number of honest/correct peers, and malicious peers that will arrive. We examined (both through simulation and experimentally) a range of values for I, C, and M to assess parameter sensitivity. Not only the scale of peers has been varied (from hundreds to thousands of peers), but also the proportion among the values assigned for I, C, and M. Since the results observed were consistent for all combinations of parameters and due to space constraints, we chose to describe (and present the results obtained for) an intermediate-size scenario, with I = 20 and C = 500.

The proposed mechanism employs a voting-based reputation scheme to classify contents as being polluted or non-polluted. Such schemes are subject to collusion attacks, in which a malicious user lies to manipulate the reputation of a given content. Hence, in both polluted and non-polluted scenarios, the attacker counts with a set of M colluding peers to attempt to defeat the mechanism. They will not only redistribute polluted content, but also provide false feedback. In the context of this work, an attacker votes positively to a polluted content and negatively to a non-polluted one. For the experimental setup being presented, the value of M was varied in the interval [0; 150] (note that these malicious users will need to have accounting the community). First, the swarm is created with I initial seeders. To evaluate the most damaging scenario, M malicious peers join immediately afterwards and contact the tracker. Peers leave the swarm only after completion of downloads, obeying a minimal contribution metric, denoted as ρ. The value of ρ represents the amount of data a peer is willing to upload in regard to what he/she has downloaded. For example, if $\rho = 0$, a peer will leave the swarm right after completing the download, regardless of the amount he/she has uploaded. If $\rho = 2$, it means a peer will attempt to upload at least twice as much content as he/she downloaded – note that it may be less than twice, when there are no interested leechers, and it can be more than twice if this amount had been uploaded before the download was completed. In this context, we define the behavior of honest and malicious peers when retrieving polluted and non-polluted contents. There are four possible cases, as follows. When a honest peer completes the download of a polluted content, we assume it immediately detects the pollution and therefore leaves the system ($\rho = 0$). Likewise, if a malicious peer finishes retrieving a non-polluted content, it does not disseminate it ($\rho = 0$). In contrast, when a malicious peer retrieves a polluted content, it remains in the swarm contributing indefinitely to its dissemination ($\rho = \infty$). The fourth and last case corresponds to the 'normal case': an honest peer retrieves a non-polluted content. The levels of collaboration in BitTorrent communities have been up to debate, so it is hard to assign 'real' values of ρ to peers. After (fully) downloading the content, 25% of leechers leave the swarm without becoming seeders ($\rho = 0$), 41% are willing to upload as much as they have downloaded ($\rho = 1$), and 34% attempt to upload twice as much as they have downloaded ($\rho = 2$). To keep the campaign of experiments manageable, we chose relatively small torrents: 60MB. The set of BitTorrent agents – the standard implementation of BitTornado – was distributed among 10 dedicated machines, interconnected through a 100Mbps network switch. Agent configuration included a limit of 7 upload connections and infinite download connections (in larger settings, the operating system would have forced a limit on this value). In addition, we adjusted the upload and download rates of the agents to 256Kbps and 1Mbps, respectively. Finally, we have assigned the following values for FUNNEL input parameters: Amin = 1, Afree = 50, a = 0.5, and σ = 0.95. These values were chosen because they represent a reasonable balance between a more conservative, rigid dissemination policy and a more relaxed one.

Effectiveness of the mechanism

To measure the effectiveness in both polluted and non polluted scenarios, we measured how long it took to each honest peer to complete its download. Figure 3 illustrates the amount of time peers stay online (up to completion) in both non-polluted and polluted scenarios (respectively, Figure 3(a) and Figure 3(b)). A pair (x,y) in the plot means that a fraction of peers x stay online for up to y minutes in order to complete their download (b) Polluted content

Fig. 3 Effectiveness against pollution attacks with colluding peers

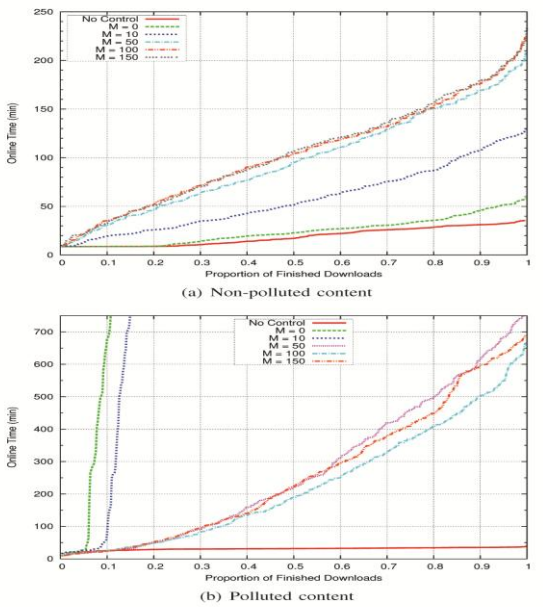

Fig. 3. Effectiveness against pollution attacks with colluding peers

Figure 3(a) illustrates the behavior of the mechanism for non-polluted scenario. In this case, the lower the curves, the better. First, note that the curves 'No Control' and 'M = 0' are similar. That is, when there are no attackers (M = 0), the presence of the pollution control mechanism introduces a negligible overhead. More precisely, results indicate that the delays imposed by the mechanism are not larger than 10 minutes to 80% of peers.

When running an experiment with 10 malicious peers, the mechanism receives votes against the legitimacy of the content (negative), which induces FUNNEL to slow down the dissemination of non-polluted content. This may be observed by the fact that honest peers took at most 70 minutes to download the content, in comparison to the curve 'M = 0'. However, we note that, to duplicate this delay, it takes five times more attackers (M = 50). In addition, the efficiency of the attack decreases as the number of malicious peers exceeds 50. Such phenomenon occurs because when colluding peers arrive, at most $A = 0.5 \times (1 + 50) + 1 = 26.5$ of the M peers are allowed to start downloads ($E[\omega] = 0.5$ since in the initial stage there are no votes recorded, thus $E[\omega] = a$). Therefore only $M - A$ malicious peers may "compete" with the honest peers in the swarm for an opportunity to start the download. Results regarding the polluted scenario are shown in Figure 3(b). In the absence of control, 100% of peers complete the download of the polluted content in less than 50 minutes. In this case, the benefits of using FUNNEL are highly expressive: approximately 90% of peers consume more than 750 minutes (12.5 hours) to complete the download of the polluted content (curve 'M = 0'). Although the plot was adjusted for the sake of presentation, the curves corresponding to cases M = 0 and M = 10 were observed for approximately 15 hours, and their behavior were stable. The mechanism also performs efficiently, controlling the dissemination of polluted content, even in the presence of 10 malicious peers voting positively. When $M \geq 50$, the set of malicious users obtained success in promoting content dissemination; however, note that to obtain the 60MB file, it took the honest peers approximately 600 minutes (10 hours), 12 times more than in the absence of FUNNEL. We emphasize that, in the experiments carried out to evaluate our proposal, peers do not give up on trying obtaining a non-polluted content. In practice, peers may leave the network due to the delay to start the download, potentially making these curves increase faster.

IV. CONCLUSIONS

BitTorrent-based file sharing communities are very popular nowadays. Anedoctal evidence hints that such communities are exposed to content pollution attacks (i.e., publication of 'false' files, viruses, or other malware), requiring a moderation effort from their administrators. The size of such a cumbersome task increases with content publishing rate. To tackle this problem, we propose a generic pollution control strategy and instantiate it as a mechanism for BitTorrent communities. The strategy follows a *conservative* approach: it regards newly published content as polluted, and allows the dissemination rate to increase according to the proportion of positive feedback issued about the content. In contrast to related approaches, the strategy and mechanism avoid the problem of pollution dissemination at the initial stages of a swarm, when insufficient feedback is available to form a reputation about the content.

This paper proposes and analyzes a pollution control strategy based on the limitation of the number of simultaneous downloads. The dissemination rate is controlled according to the trust in the version correctness. Initially, the mechanism is proposed and evaluated using ideal settings. Simulation results showed the effectiveness of the mechanism in containing pollution. Later, we identify the main design decisions, and then propose four basic distributed mechanisms based on classic designs. The mechanisms are compared in terms of overhead (delays imposed on the dissemination of non-polluted content) and efficiency in reducing the dissemination of polluted versions, with and without the help of collusion attacks. Solving the pollution problem is a challenging issue which will require many steps. The main contribution of this paper is to propose a strategy and related mechanisms for pollution control, based on a reputation scheme built around Subjective Logic. Instead of a system, we introduce a novel approach in the design space of pollution control mechanisms, which may influence the design and implementation of novel, more secure P2P file sharing systems in the future.

REFERENCES

[1] Christin N. Weigend AS. Chuang J. "Content availability, pollution and poisoning in file sharing peer to peer network",Sixth ACM Conference on Electronic Commerce(EC 05),68-77.

[2] Liang J. Kumar R.Xi Y ,Ross K. "pollution in p2p file sharing system", Twenty fourth IEEE international Conference on computer communications(INFOCOM 2005) , vol. 2, 1174-1185.

[3] Liang J. Naoumov N ,Ross KW. "The index poisoning attack in p2p file sharing system", Twenty fifth IEEE international Conference on computer communications (INFOCOM 2006) ,1-12.

[4] Costa C Almeida J. Almeida V " Fighting pollution dissemination in peer to peer file sharing network", Seventh IEEE International Conference On peer to peer Computing (p2p 2007), 53-60.

[5] Liang J. Naoumov N ,Ross K. "Efficient blacklisting and pollution level estimation in p2p file sharing ", Technology For Advanced Heterogeneous Network(Lecture Notes in Computer Science Vol.3837 .Springer Dec 2005,1-21

[6] Lee U Choiz M, Choy J. Sanadidiy MY Gerla M. " Understanding pollution dynamics in p2p in p2p file sharing ",Fifth International Workshop on peer to peer System IPTPS 06) Feb 2006, 1-6

[7] Kumar R. Yao DD Bagehi A Ross KW , Rubenstein D. " Fluid modeling of pollution proliferation in p2o network" ACM SIGMETRICS Performance Evaluation Review 2006,335-346.

[8] Thommes R.Coates M Epidemiological "modeling of peer to peer viruses and pollution"Twenty Fifth IEEE international Conference on computer communications (INFOCOM 2006),981-993.

[9] Walsh K, Sirer EG , "Fighting peer to peer spam and decoys with object reputation ", Third ACM SIGCOMM Workshop on the Economics of peer tp peer Systems(p2pecon 05),138-143.

[10] Walsh K, Sirer EG , "Experience With a Distributed Object Reputation System For peer to peer filesharing ", Third USENIX Symposium on Networked System Design and Implementation (NSDI 06) May 2006,1-14.

[11] Costa C Soares Almeida J. Almeida V " Fighting pollution dissemination in peer to peer file sharing network", Twenty second Annual ACM Symposium on Applied Computing (SAC 07)1586-1590

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in 12th InternationalConference on World Wide Web (WWW 2003). NewYork,NY, USA: ACM, 2003, pp. 640–651.

[13] A. Josang, S. Pope, and R. Hayward, "Trust network analysis with subjective logic," in 29th Australasian Computer Science Conference (ACSC 2006). Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 85–94.