Future Network and MobileSummit 2010 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2010
ISBN: 978-1-905824-16-8

Information Ranking in Content-Centric Networks

Nikos FOTIOU, Giannis F. MARIAS, George C. POLYZOS
Mobile Multimedia Laboratory, Athens University of Economics and Business
Patision 76, Athens, 113 62 Greece
Tel: +30 210 8203 693, Fax: + 30 210 8203686
Email:fotiou@aueb.gr, marias@aueb.gr, polyzos@aueb.gr

Abstract: Content-centric networking is often regarded as a promising paradigm for future networks. Nevertheless current content-based networks–such as file sharing peer-to-peer (P2P) networks–have been proved to be vulnerable to content pollution attacks. A significant amount of research efforts has been launched in order to mitigate this kind of attacks. The majority of these efforts focuses on users' ranking, based on their behavior, while little work has been done in ranking information itself. We show in this paper that solutions based on users' ranking can be by-passed by malicious users. Furthermore we propose *inforanking*, a light-weight solution for ranking information, based exclusively on positive votes. We compare our solution to Credence object-based reputation system. Our solution demonstrates significant less burden to the network and outperforms Credence in terms of polluted content isolation.

Keywords: Content-based networks, Content pollution, Information ranking

1. Introduction

Content-based networking has been in the spotlight of Future Internet research efforts–such as CCNx [1] PSIRP [2] 4WARD [3]. Its information-centric nature swifts focus from end-users to information itself opening the ground for new innovating applications. Nevertheless content-based networks do currently exist and the security issues they face are a useful guide for the security solutions that should be developed for the Future Internet. One of the most popular category of content-based networks is p2p file-sharing networks. File-sharing networks have been widely deployed allowing users for searching content–such as songs and videos. In these networks requests for content trigger, in a transparent way, a series of protocols for content location and transfer. End-users are unaware of all underlayed protocols and infrastructure, and their main concern is to describe in a proper way the desired content.

However, file-sharing networks suffer from content pollution attacks. Being mainly used for illegally exchanging intellectually property products, file-sharing networks cause a big income loss to content industry, which in order to protect its products pollutes these network. Moreover being used by thousands of users, file-sharing networks are usually the playground of virus and worms developers. In fact Liang et al. [4] found out that in Kazaa file sharing network there existed more than 20.000 versions of some popular files and more than 50% of them were polluted. The polluted versions, in some cases, corresponded for more than the 60% of the total copies of the file. A similar research [5] showed that in FastTrack filesharing network for some popular items, about the 70% of their copies and the 60% of their versions were polluted. Kalafut et al. [6] measured that, 68% of all downloadable responses in Limewire containing archives and executables, were actually malware. Shin et al [7] reported that in KaZaA network

in response to 24 common query strings over 15% of the results were infected by 52 different viruses.

This paper presents inforanking, a light-weight solution for ranking information, eventually leading to the isolation of polluted pieces of information. Inforanking tries not to impose any extra overhead to the overall network and it takes advantage of the functionality already provided by the information-sharing networks. Inforanking is the result of the observation that malicious users, provide numerous polluted versions, in order to avoid blacklisting. Our suggested solution allows users to give positive votes only, and it weights users' votes reversely proportionally to the number of versions they provide.

The rest of this paper is organized as follows. Section 2 presents related work in the area, emphasizing Credense object-based reputation system, Section 3 demonstrates the Byzantine users attack and it presents the inforanking solution which is evaluated in Section 4. Finally Section 5 presents our conclusions and future work.

2. Related Work

Various proposals for ranking users exist in the literature. Eigentrust [8] is a popular algorithm for reputation management in p2p networks which enables peers to express their trust on other peers based on their (un)satisfaction, and it provides mechanisms that allow for local trust transition, correlation and aggregation. Scrubber [9] in a similar manner tries to rank users based on their behavior, while its extension [10] uses a hybrid approach of both users and objects ranking. We show in section 3.1 how can Byzantine users affect the performance of solutions based on user ranking.

Credence [11], on the other hand, is an object reputation approach for fighting content pollution in p2p file sharing systems. We analyze further Credence in section 2.1 and we compare it with inforanking in the Evaluation section. PageRank [12] is an algorithm for ranking webpages. The incoming links towards a page are considered as positive votes. Each link is weighted in relation to the rank of the page of origin as well as to the number of outgoing links of the page of origin. There does not exist negative vote in PageRank. Inforanking borrows the positive votes only approach of PageRank and it extends it to a content-specific context, i.e., while in PageRank each URL/URI has its own fixed rank in Inforanking the rank of an information item depends on the context this item has been requested. LIP [13] uses file's average retention time in users' computers in order to detect fake files, nevertheless it generates false positives reports if files have been simply renamed.

2.1 The Credence Object-Based Reputation System

Credence is a weighted voting protocol in which a peer may vote positively $(+1)$ or negatively (-1) on any object regarding its authenticity. A positive vote is interpreted as an indication that the object's description matches the object's content whereas a negative vote indicates the opposite, i.e, the object's description and its content differ.

Credence voting protocol works as follows. Any peer 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 peer that posses votes responses. The collected votes are weighted using a weighting factor r. This factor reflects the relationship between two peers and it takes values in the range $[-1, 1]$.

The relationship between two peers, A and B, is expressed by the coefficient of correlation of their voting histories and it is computed as $\theta = (p-ab)/\sqrt{a(1-a)b(1-b)}$ where a and b is the fraction where A and B voted positively, respectively, and p the fraction where both A and B voted positively. The vote weighting factor r equals to θ when $|\theta| >= 0.5$, i.e, when the two peers tend to (dis)agree on the objects they vote positively, and to 0 when $|\theta| < 0.5$,i.e., when the two peers have uncorrelated voting history. For peers which have voted only positively or negatively θ is undefined. In that case a vote agreement metric is used with maximum $|r| = 0.75$.

Each peer maintains a local vote database where the gathered votes are stored. These databases is used to answer incoming vote-gather queries as well as to calculate correlations with other peers. Moreover each peer maintains a list of peers with which is highly correlated. This list is periodically exchanged between highly correlated peers so as transitive correlations to be created. Transitive correlation reflects the notion that if A and B are highly correlated and B and C are also highly correlated then there should also exist a correlation between A and C which is calculated as $\theta_{AC} = \theta_{AB} * \theta_{BC}$

3. Approach

3.1 Why an Information Centric Approach?

It is argued that we are moving towards an information-centric Internet [14] [15] which will provide us with even more functionality regarding information manipulation. Moreover it can be proved that even with the current standards end-users rating is not the best solution in terms of malicious information isolation. This happens due to the fact that users change behavior and a trustful user may suddenly start behaving maliciously. In order to demonstrate that, we have simulated a small p2p file-sharing network where end-hosts are rated using a variation of EigenTrust [8] algorithm. In this network there exist 10 information providers, sharing the same 10 pieces on information. 50% of them, i.e., 5 providers, are malicious and when requested for the 10^{th} information item they send bogus data. However the malicious providers behave in a normal way when they are requested any other item. 100 users enter the network and they start requesting the 10 items in a random order. Every time they receive an item they vote positively or negatively the item provider. A positive vote is interpreted as the result of a satisfactory transaction whilst a negative vote as an unsatisfactory one. We assume that voters never lie and every vote is stored safely in a centralized storage accessible, by everybody. Thus every vote is treated by a user as being the result of a personal transaction. As a result we modify EigenTrust as follows: The local trust value of peer i toward j is calculated using the following formula $s_{ij} = sat(*, j) - unsat(*, j)$ where $(un)sat(*.j)$ are all the (un)satisfactory votes for j. The local trust value is normalized as follows $c_{ij} = max(s_{ij}, 0)/max(\sum_j s_{ij}, 0)$. Because of our assumptions there is no need for local trust values transition and aggregation, as local trust values are calculated using everybody's–real–votes. Figure 1 shows the percentage of bad downloads for queries concerning information item 10. As it can been seen user rating did not manage to prevent bad downloads as their percentage is above 40% of the total downloads concerning this specific item. The fact that generally well-behaving users can spread bad content when suddenly start misbehaving may drive malicious users start attacking them in order to manipulate them.

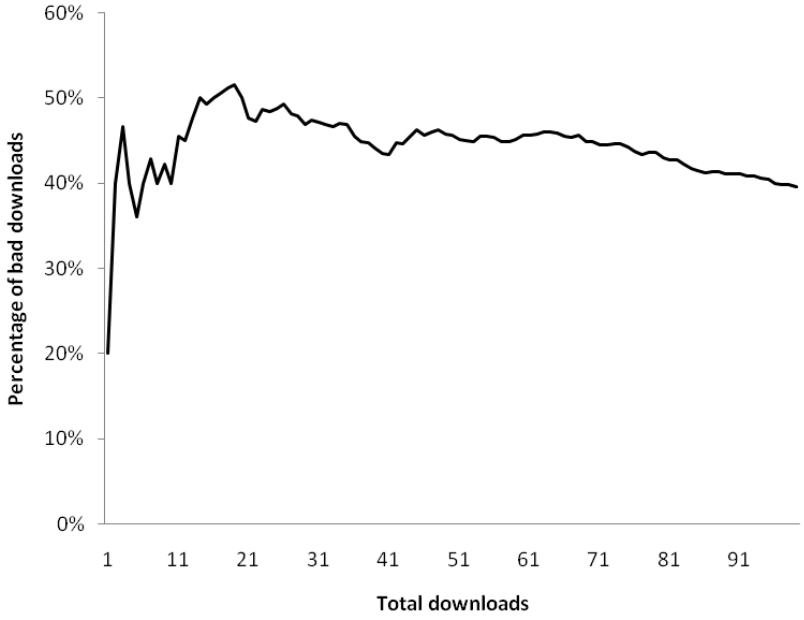

Figure 1: User rating approach evaluation

3.2 Reference Architecture

In our approach we consider a system in which users are interested in a specific piece of information. This piece of information may exist in various versions and each version can be provided by numerous users. A version can be valid or polluted. A valid version contains the desired piece of information whilst the polluted does not. Moreover the decision whether a version is polluted or not is based exclusively on objective criteria. All users advertise a description of the various pieces of information they provide along with an identity for each piece, which for example can be the result of a hash function over the information's data. Users sharing the same information piece advertise the same–or similar–description and the same identity. An entity, which can be distributed or centralized, collects these advertisements and maintains a database of descriptions, identities and providers. Every user can query this entity using keywords regarding a specific piece of information. The result of a user's query is a list of information identities, whose description matches the user's keywords, as well as a list of providers for each identity. Our target is to provide a tool which will enable users to distinguish and isolate polluted versions. Every user is identified by a unique identifier. We assume that it is difficult for a user to generate multiple identifiers as well as to use an identifier that does not belong to him. Our reference architecture bares high resemblance to a file sharing network.

3.3 Voting

Users may vote only positively regarding a specific information version. Moreover a user may vote only once for a specific information version. In order to avoid any additional network traffic or state maintenance, we may regard the fact that a user shares an information version as a positive vote, i.e., every user gives one positive vote to each file version he shares. When a query for a specific information item is performed a result list, containing version's identifiers and providers is created. Every version identifier

VersionID	Providers	Score
VER01	PRO1, PRO2, PRO3, PRO4	1.75
VER02	PRO1, PRO2, PRO3, PRO4	1.75
VER03	PRO1, PRO5, PRO6, PRO7	3.25
VER04	PRO1	0.25

Table 1: Voting example

has as many positive votes as the number of its providers. Nevertheless not all votes are equal. Each vote of a provider ***P*** in a result set ***R*** is weighted by a factor w computed as $w = 1/(\sum P_R)^a$ where $\sum P_R$ is the sum of P's votes in R and a a fixed value. As an example consider Table 1. Provider PRO1 has voted for 4 items in the result set, so his vote is weighted by 0.25, On the other hand providers PRO5, PRO6 and PRO7 have voted only once so their votes are weighted by 1

4. Evaluation

We evaluate inforanking through the simulation of a high polluted environment and we compare it against a naive solution as well as against Credence object-based reputation system. We focus on malicious object isolation. We do not consider network or storage overhead as we believe that is negligible in our proposed solution.

4.1 Simulation Setup

Using OMNeT++ [16] and OverSim framework [17], we simulate a network consisting of 100 users 30 of which are malicious. Moreover we consider 10 information items with 20 versions each. 12 out of the 20 versions are polluted, containing malicious content. In each simulation round there exists a warming up period during which users select the objects that will provide. Each malicious user selects 20 objects, while each other user selects 5 objects. When all users have selected their objects, the non-malicious ones start querying the network in order to obtain a valid version of all the other files they do not have. Every time a user downloads a bad version in the next round it retries to download a valid version of the same information item. We simulate and evaluate 3 different strategies. All non-malicious users follow the same strategy. The first strategy is the ***naive*** strategy. When using naive strategy, users download the version with the most positive votes. If the downloaded file is a valid one, the user gives a positive vote otherwise he votes negatively. The second strategy is the Credence object-based reputation system and the third strategy is inforanking. We also assume that when using the Credence strategy users always vote, and they vote correctly, malicious users vote negatively for valid objects and when θ is undefined $r = 0.5$. Moreover when our solution is used, non-malicious users never share a polluted object. Each simulation round lasts until all non-malicious users download all the valid information items. Moreover each experiment is repeated 5 times.

Figure 2 shows that even in this high polluted environment our approach converges much faster than the other two solutions. It can been easily shown that when malicious users share in average V_m versions and non-malicious V_g of an information item, and there exist U_g non-malicious users in the network the number of malicious users should be $U_m = (V_m/V_g)^a * U_g + 1$ in order to achieve 1 download of malicious content. Moreover in order for malicious users to be successful and avoid blacklisting it should

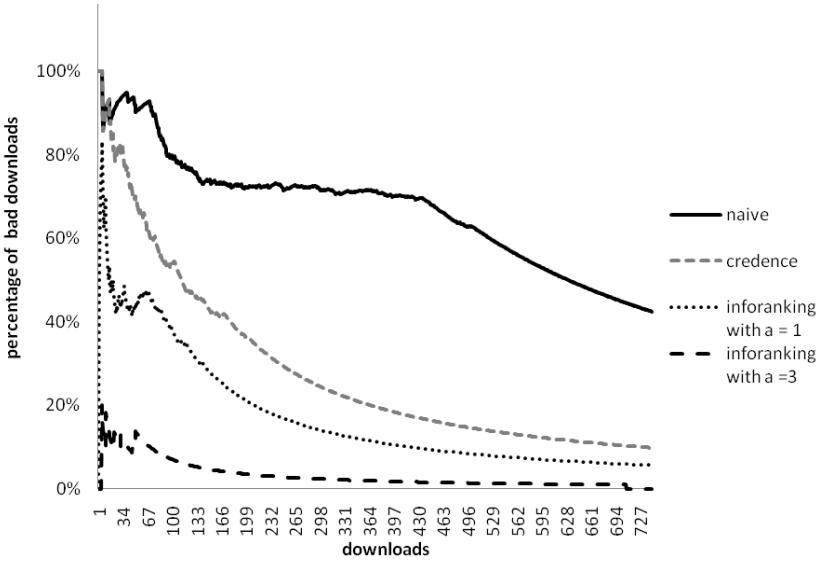

Figure 2: Percentage of bad downloads

be $V_m >> V_g$, therefore it is difficult for malicious users to achieve their goal when inforanking is used.

5. Conclusions and Future Work

In this paper we presented inforanking, a light-weight information ranking mechanism. By taking advantage of the existing functionality of content-based networks we managed to isolate polluted information items without imposing any overhead to the system. We compared our solution to Credence object reputation system and we proved that inforanking is more effective in terms of how fast a bogus item is identified and isolated.

We believe that Inforanking can be applied in file sharing networks for isolating malicious files, in voting systems for eliminating the effect of faulty votes as well as in information searching engines for finding the pieces of informations that correspond with the user's search criteria. Future work includes inforanking evaluation in real environments, including p2p file sharing networks and bittorent systems. Moreover we anticipate to research inforanking in a more general purpose content-based network.

References

[1] "Ccnx project," March 2010. `http://www.ccnx.org`.

[2] "Psirp project," March 2010. `http://www.psirp.org`.

[3] "4ward project," March 2010. `http://www.4ward-project.eu`.

[4] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in p2p file sharing systems," in *INFOCOM 24th IEEE International Conference on Computer Communications*, pp. 1174–1185, IEEE, 2005.

[5] J. Liang, N. Naoumov, and K. W. Ross, "The index poisoning attack in p2p file sharing systems," in *INFOCOM 25th IEEE International Conference on Computer Communications*, pp. 1–12, 2006.

[6] A. Kalafut, A. Acharya, and M. Gupta, "A study of malware in peer-to-peer networks," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 327–332, ACM, 2006.

[7] S. Shin, J. Jung, and H. Balakrishnan, "Malware prevalence in the kazaa file-sharing network," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 333–338, ACM, 2006.

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *WWW '03: Proceedings of the 12th international conference on World Wide Web*, (New York, NY, USA), pp. 640–651, ACM, 2003.

[9] C. Costa, V. Soares, J. Almeida, and V. Almeida, "Fighting pollution dissemination in peer-to-peer networks," in *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, (New York, NY, USA), pp. 1586–1590, ACM, 2007.

[10] C. Costa and J. Almeida, "Reputation systems for fighting pollution in peer-to-peer file sharing systems," in *P2P '07: Proceedings of the Seventh IEEE International Conference on Peer-to-Peer Computing*, (Washington, DC, USA), pp. 53–60, IEEE Computer Society, 2007.

[11] K. Walsh and E. G. Sirer, "Fighting peer-to-peer spam and decoys with object reputation," in *P2PECON '05: Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, (New York, NY, USA), pp. 138–143, ACM, 2005.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," tech. rep., Stanford Digital Library Technologies Project, 1998.

[13] Q. Feng and Y. Dai, "Lip: A lifetime and popularity based ranking approach to filter out fake files in p2p file sharing systems," *Proc. of IPTPS, Bellevue, Washington*, 2007.

[14] V. Jacobson, "If a clean slate is the solution what was the problem," Stanford Clean Slate Seminar, 2006.

[15] C. Esteve, F. Verdi, and M. Magalhães, "Towards a new generation of information-oriented internetworking architectures," in *Proceedings of the 2008 ACM CoNEXT Conference*, ACM New York, NY, USA, 2008.

[16] "Omnet++ community site," March 2010. `http://www.omnetpp.org`.

[17] "The oversim p2p simulator," March 2010. `http://www.oversim.org`.

