Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186–5140
Volume 4, Number 1, pages 27–33, January 2015

Index Poisoning Scheme for P2P File Sharing Systems with Low Spatial and Network Costs (Preliminary Version)

Yuusuke OOKITA[1] and Satoshi FUJITA[1]
[1]Department of Information Engineering, Hiroshima University
Higashi-Hiroshima, 739-8527, Japan

***Abstract*—In this paper, we propose an index poisoning scheme for unstructured P2P file sharing systems. Index poisoning is a technique which alters the index of illegal files so that they could not be reached by any peer which wishes to access them through the P2P. The proposed scheme is a combination of the following three periodical injections of the copies of altered index: 1) injection to the owner of illegal files, 2) injection to upstream peers by setting a long lifetime to each copy, and 3) injection to downstream peers by setting a short lifetime to each copy. The results of simulation indicate that the proposed scheme attains almost the same performance with a simple index poisoning scheme which injects altered copy with long lifetime to all peers, with a significantly lower cost, i.e., the spatial cost reduces to one third and the peak network cost of external agent reduces to a half.**

***Index Terms*—Unstructured peer-to-peer, file sharing, index poisoning, illegal file sharing.**

I. Introduction

Peer-to-Peer (P2P) technology has been widely used in many fields including IP-phone, live streaming, and file sharing [9], [10], [13]. In particular, P2P file sharing is recognized as a key application for the P2P technology, and there are many P2P file sharing systems used by many users from the late 90's to the present, which include Napster, Gnutella, Kazaa and BitTorrent. However, the popularization of P2P file sharing software causes several critical issues in recent years, such as the illegal sharing of copyrighted contents violating the copyright law and the act as a hotbed of cyber-crimes such as phishing scams and leaks of personal information infected by malware hidden in illegal files.

There are many challenges conducted so far to restrain such downloads of illegal files by anonymous users. DRM (Digital Rights Management) is a technology which encodes contents using a specific encoding technique so that it could be decoded merely by using specific software and/or hardware [3]. An example of DRM is the technology used in Windows Media DRM[1], which requests private key for playing back the encoded content which is individually issued for each paid customer. The way of detecting the occurrence of illegal file sharing has also been investigated extensively. For example, we could identify the user who illegally leaks paid contents to other unauthorized users with the aid of digital watermarking [3] and we could identify a group of users who illegally share paid contents by deploying a decoy peer in the group of colluding users [1], [2], [7].

Among them, index poisoning has recently attracted considerable attention by many researchers [4], [6], [8], [11]. Index poisoning is a technique which alters the index of illegal files so that they could not be reached by any peer which wishes to access them through the P2P [6]. In pure P2P systems which do not rely on a specific index server, copies of the **index** of files, which contains the name, owner, size and the other attributes of the files are distributed over the network beforehand, so that it can be efficiently retrieved by propagating a query message over the network. More precisely, if a peer which receives a query from its neighbor has a copy of an index matching the query, it replies the index to the questioner so that the body of the file can be accessed by referring to the received index. The received copy can also be held by the questioner, while the number of copies which can be held by each peer is limited by a constant. In addition, each copy is given a lifetime and is periodically updated by the file owner, so that the index of popular files will be held by many peers, where the way of distributing and retrieving indices depends on the underlying file sharing software.

This paper was supported by the Grant-in-Aid for Scientific Research No.24500082 of the JSPS.

[1]http://msdn.microsoft.com/library/cc838192(v=vs.95).aspx

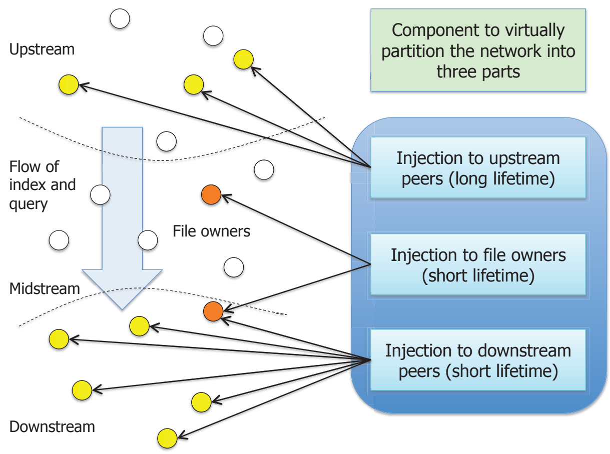


Fig. 1. Overview of the proposed scheme.

By altering the index of illegal files and by propagating it to *all* peers in the system with a sufficiently *long* lifetime, we can significantly reduce the probability of correctly identifying the location of illegal files. However, such an unlimited propagation of long lived copies would cause another problem so that the most of copies held by the peers are altered ones. Although there are many index poisoning schemes proposed in the literature [4], [6], [8], [11], to the authors' best knowledge, none of them could successfully balance the cost and the performance, as will be briefly reviewed in Section IV.

In this paper, we propose an index poisoning scheme for unstructured P2P file sharing systems. The proposed scheme consists of the following four components:

1) The first component identifies the direction of the flow of indices and virtually partitions the set of peers into three parts as upstream peers, midstream peers, and downstream peers;
2) The second component periodically injects copies of altered indices with short lifetime to the *owner of illegal files* with time interval T_1;
3) The third component periodically injects copies of altered indices with long lifetime to *upstream peers* with time interval T_2; and
4) The fourth component periodically injects copies of altered indices with short lifetime to *downstream peers* with time interval T_3.

See Figure 1 for illustration. The reader should note that in the proposed scheme, altered copies are injected from the outside of the P2P network by an agent such as the system manager and the security organization. By combining three different injections with an appropriate tuning of time intervals and the lifetime, we could reduce the cost of index poisoning by keeping the probability of identifying the location of illegal files sufficiently low. The proposed scheme is evaluated by simulation. The simulation results indicate that the proposed scheme attains almost the same performance with a simple index poisoning scheme which injects altered copy with long lifetime to all peers, with a significantly lower cost, i.e., the spatial cost reduces to one third and the peak network cost of external agent reduces to a half.

The remainder of this paper is organized as follows. Section II describes the model of P2P systems considered in this paper including the cost model and the performance metrics. Section III describes the details of the proposed scheme. Section IV overviews related works. Section V shows the result of simulations. Finally, Section VI concludes the paper with future work.

II. Model

This paper considers a P2P file sharing system based on an unstructured P2P. The index of each shared file is distributed over the network using an appropriate distribution protocol so that it could be retrieved by propagating a query through the network. The sharing of illegal files is disturbed by applying an index poisoning scheme. Under index poisoning schemes, the probability of successfully identifying the (correct) location of illegal files is a function of the following factors: 1) the way of propagating the query to the network; 2) the way of distributing index to the network; 3) the number of correct copies of the index of illegal files; and 4) the ratio of the number of copies of correct index to the number of copies including altered and correct ones. The goal of index poisoning schemes is to reduce the probability of correct identification as much as possible by reducing the third and the fourth factors described above, while keeping the required cost sufficiently low.

As for the metrics for the cost of index poisoning schemes, we focus on the following two factors:

- **Network cost** which is evaluated by the total number of messages as well as the load of agent to inject altered copies to the network; and
- **Spatial cost** which is evaluated by the number of altered copies existing in the network.

To reduce the network cost, we should design a scheme so that altered copy will reach peers holding the correct copy in a distributed manner, and to reduce the spatial cost, we should design a scheme so that the lifetime of each copy is determined in such a way that altered copy immediately disappears after arriving at the target peer.

III. Proposed Scheme

A. Outline

As was described in Section I, the proposed scheme is a combination of the following three periodical injections of copies of altered index:

1) Injection to the *owner of illegal files* with a short lifetime which is aimed to remove the "seed" of correct copies from the network.
2) Injection to *upstream peers* with a long lifetime which is aimed to be propagated autonomously through the underlying overlay; and
3) Injection to *downstream peers* with a short lifetime which is aimed to remove copies from the network by overwriting propagated ones.

In the following subsections, we describe the concrete way of each injection in detail.

B. Injection to Remove the Seed of Propagation

The first injection is targeted at the file owner associated with the altered index. By overwriting a copy of the index held by the file owner with an altered one, we can stop the new propagation of correct index starting from the file owner, until a copy of correct index is generated by itself or is received from adjacent peers. The lifetime of injected copy can be small, since the role of this injection is merely to stop the new propagation, and the removal of propagated (correct) copies will be done by the combination of the next two injections. The location of the file owner can be identified by observing the TTL (Time To Live) of messages containing the copy of index [11], since the message transmitted by the owner should have the largest TTL among others. One possible way to realize such an observation is to insert a dummy (decoy) peer to the network which gradually changes its position in the network toward the direction of the originator, and another way is to deploy a crawler which crawls TTL of messages transmitted in the network [11].

C. Injections to Remove Propagated Copies

The aim of the second and the third injections is to remove correct copies which have been distributed over the network. To resolve the trade-off between the network cost and the spatial cost, we take an approach so that the role of spreading altered index to as many peers as possible is separated from the role of removing spread copies as soon as possible. More concretely, in the second injection, we inject copies of altered index to upstream peers by setting a long lifetime so that it will be received by many peers which might or might not hold a copy of correct index, and in the third injection, we inject copies of altered index to downstream peers by setting a short lifetime so that it overwrites altered index received through the second injection and is removed from the network within a short time. Although copies of altered index received by midstream peers will not be removed by the third injection, it does not matter in our scheme since those copies continuously serve as a seed for downstream peers until the lifetime exhausts.

The identification of upstream and downstream peers can be realized by observing the behavior of each peer concerned with the index propagation. Another way, which is specific to Winny, is to trace paths traversed by query messages [8]. More concretely, Winny is designed in such a way that each query message is forwarded toward the downstream to increase the hitting rate as much as possible, and that it records the set of peers on the traversed path to avoid redundant message transmissions. Thus by referring to the paths traversed by queries, we can identify upstream peers which frequently appear at the beginning of the paths, and downstream peers which frequently appear at the end of the paths [8].

IV. Related Work

Before proceeding to the evaluation of the proposed scheme, we will briefly overview related works. Liang *et al.* [5] proposed a technique called **pollution attack** which is aimed to prevent users from conducting an illegal sharing of copyrighted contents in FastTrack. FastTrack is a P2P file sharing system widely used in early 2000s which has a hierarchical structure consisting of ordinary peers (OPs) and super-peers (SPs). Each OP joins the (hierarchical) P2P as a child of an SP and the index of files held by the OP is uploaded to the parent SP. Query message issued by a peer p is propagated through the overlay network consisting of SPs, and after arriving at an SP q holding an index matching the query, a reply message including the index is directly returned from q to p. The basic idea of the pollution attack is to intentionally contaminate illegal files, and to propagate such a contamination to all peers holding a correct copy so that no peer can access uncontaminated copy of illegal files. This approach works well in FastTrack since the hash function used in FastTrack does not fully check the authenticity of acquired files. However, it does *not* work well in other P2P systems such as BitTorrent, since in those systems, a hash function such as SHA-1 is applied to the whole of every piece derived from the given file, and all of the resulting hash values are written in a file (e.g., .torrent file in BitTorrent) which needs to be acquired before starting the download so that the authenticity of acquired pieces is checked by referring to the hash values.

To overcome such a weakness of the pollution attack, Liang *et al.* proposed another technique called **index poisoning** [6]. As was described previously, the key idea of the index poisoning is to intentionally alter the index of files to reduce the probability of successfully downloading illegal files. It was demonstrated that such a technique is effective not only in unstructured P2Ps such as FastTrack but also in DHT-based structured P2Ps such as Overnet. The concrete index poisoning procedure for FastTrack proposed by Liang *et al.* proceeds as follows: 1) the agent establishes a TCP connection to all SPs holding the index of illegal files, and 2) it alters the value of either ID, IP address, or the port number contained in the index. Although it is possible to alter all indices of illegal files by this approach, it would take a long time to establish TCP connection to all SPs holding the index of illegal files since FastTrack accommodates more than 20 thousands SPs in 2006. The impact of index poisoning in BitTorrent is verified in [4] with experimental results in actual P2P systems.

Yoshida *et al.* [11] proposed an index poisoning scheme for Winny, which is a P2P file sharing system widely used in Japan. This scheme involves a mechanism which *removes a copy of the index of illegal files immediately after overwriting the correct copy*. More concretely, it uses the property of Winny protocol such that every index received from other peers overwrites a copy of the index with the same ID without verifying the authenticity, so that it can remove the overwritten copy by setting the lifetime to zero. The intention behind the Yoshida's scheme is to reduce the spatial cost as much as possible. Such an intention can be certainly attained by setting the lifetime of altered indices to be zero, but as a side effect, it should give up the "propagation" of altered indices which causes a heavy load at the centralized agent (the experimental result shows that it takes 15 min to distributed altered index to about 100000 peers [11], but apparently it does not scale). In addition, such a heavy task of the agent should be conducted periodically since in general, the correct index of illegal files could be (re)generated by the file owner.

Such a drawback of the Yoshida's scheme is partially overcome by Putra *et al.* [8] by modifying the scheme so that altered index is injected to a specific part of the network which maximizes the effect of injection. Such a modification certainly reduces the load of the agent to 28% of the Yoshida's scheme without reducing the number of peers which receive altered index, while as a side effect, we need to increase the lifetime of each copy to 1500 sec, i.e., it significantly increases the spatial cost.

V. Evaluation

A. Setup

We evaluate the performance of the proposed scheme by simulation using an open source P2P simulator PeerSim[2]. As a concrete P2P file sharing system, we focus on Winny and compare the performance of the proposed scheme with a simple (but inefficient) scheme in which altered index is injected to all peers.

In Winny, participant peers are classified into three types by the computational power so that upstream peers, midstream peers and downstream peers[3]. Parameters used in the simulation are set as follows:

- Number of peers: 1000
- Kind of files: 500
- Number of keys initially held by each peer: 2
- Maximum number of keys held by each peer: 500 for downstream peers, 300 for midstream peers and 100 for upstream peers.
- Number of peers of each type: Among 1000 participants, 235 are downstream peers, 485 are midstream peers and 280 are upstream peers [8].
- Default lifetime of index: 1500 sec.
- Simulation time: 3870 sec.

The system contains exactly one illegal file, i.e., the remaining 499 files are legal ones. This illegal file is held by one particular peer at time 0, and is randomly copied to other peers during simulation due to the transfer of the body of file; i.e., the number of *file owners* gradually increases. In all schemes, the first injection of altered index takes place when it pasts 1860 sec after the beginning of the simulation. The reader should note that since the default life time is set to 1500 sec, if the number of files owners did not increase by the time of the first injection, copies of (correct) index of the illegal file should disappear (except for the original one) just before the first injection.

Setting specific to the proposed scheme is as follows:

1) Injection to upstream peers is conducted so that about 170 upstream peers connecting to midstream peers are selected as the target of injection, and altered indices are periodically injected to such peers. The period of injection is identical to the lifetime of the altered index, which is either 150, 210, 300 or 450 sec in the simulation. The reader should note that although we do not

[2]http://peersim.sourceforge.net/

[3]In the terminology of Winny, upstream peers in the current paper are called "downstream peers" and downstream peers in the current paper are called "upstream peers." More precisely, "upstream peers" in Winny have more computational power than "downstream peers" and indices and queries are propagated from "downstream peers" to "upstream peers" as in FastTrack.

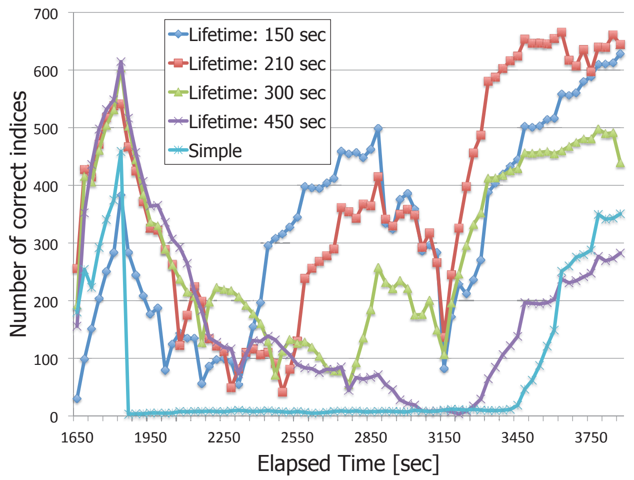


Fig. 2. Time variation of the number of correct copies; a comparison with the simple scheme.

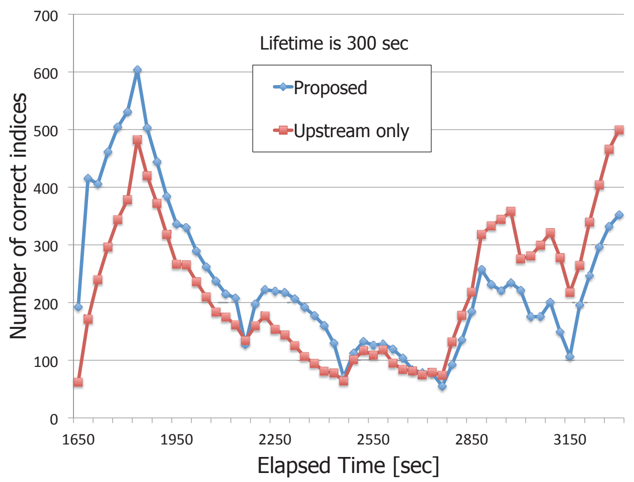


Fig. 3. The effect of the injection to downstream peers.

explicitly evaluate the cost of the proposed scheme, it is much smaller than the conventional scheme in which copies with lifetime 1500 sec are injected to all peers.

2) Injection to downstream peers and injection to file owner(s) are periodically conducted by setting the lifetime of injected copies to zero. In the experiments, we fix the period of injection to be identical to the period of injection to upstream peers.

B. Number of Copies of Correct Index

At first, we evaluate the effect of the proposed scheme to the number of correct copies of the index of illegal file. Figure 2 illustrates the time variation of the number of correct copies, where curves in the figure correspond to the simple scheme and the proposed scheme with different settings of the lifetime, respectively.

As was claimed previously, the number of copies is reset to one at 1500 sec and then rapidly increases according to the index distribution protocol of Winny. Although those copies immediately disappear after the injection under the simple scheme (at 1860 sec), under the proposed scheme, the number of copies gradually decreases by repeating vibration, where the period of vibration coincides with the period of injection. The effect of controlling the number of copies increases as the lifetime of altered index increases, and in this simulation, lifetime of 450 sec attains almost the same performance with the simple scheme at time around 3000 sec. As was mentioned previously, the simple scheme injects altered index of lifetime 1500 sec to all of 1000 peers, which is much larger than the proposed scheme in which: 1) altered index of relatively large lifetime (e.g., 450 sec) is injected to about 170 upstream peers and 2) altered index of lifetime zero is injected to 235 downstream peers and few file owners. Hence, although the proposed scheme should repeat injection with relatively short time interval, we can conclude that it significantly reduces the peak load of the external agent while keeping the effect of index poisoning sufficiently high.

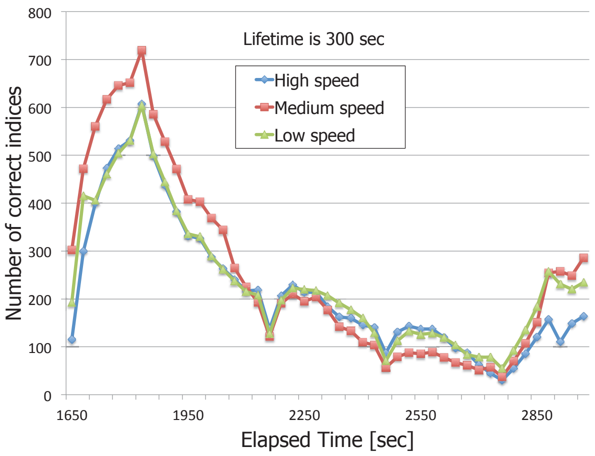
The impact of the injection to downstream peers is shown in Figure 3. In this figure, time variation of the proposed scheme is shown by blue curve and that without the injection to downstream peers is show by red curve. The number of correct copies issued at time 1500 sec rapidly increases and the first injection to upstream peers is conducted at time 1860 sec. After that, such an injection is periodically conducted with interval 300 sec. During the first three injections, red outperforms blue due to a larger number of copies existing before starting the injection, but after the fourth injection, blue outperforms red; i.e., we can claim that the injection to downstream peers is certainly effective to reduce the number of correct copies.

C. Impact of Update Interval of File Owner

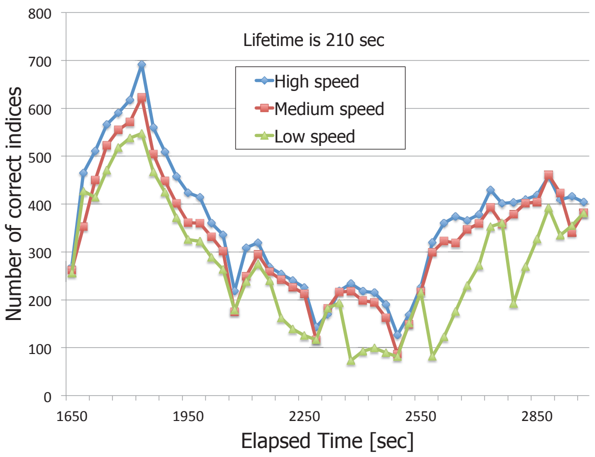
Such an effect of the proposed scheme is affected by the *update interval* of file owner. More concretely, if the update interval becomes longer, then we could attain the same performance by setting shorter lifetime. To verify this intuition, we conduct the following simulations:

- Fix the lifetime of altered index to 210 sec or 300 sec, and
- Slow down the update speed of file owner(s) to $1/2$, $1/3$, and $1/4$ of the original Winny.

Figure 4 shows the results. From the figure we can claim that the impact of the difference of update interval becomes large when the lifetime is relatively small.



(a) When lifetime is 300.



(b) When lifetime is 210.

Fig. 4. Impact of the update interval of file owner.

D. Hit Rate of Queries

Finally, we evaluate the impact of schemes to the hit rate of queries. Recall that the objective of index poisoning is to reduce the hit rate of queries requesting the (correct) index of illegal files. Figure 5 illustrates the hit rate concerned with illegal files, where the horizontal axes are the lifetime of altered index and the update interval of file owner, respectively. Although we could not find a clear relationship between the update interval and the hit rate, we can conclude that the hit rate monotonically decreases as the lifetime increases. For example, when the update interval is high, the hit rate for lifetime 450 sec is less than 0.5, while the hit rate for lifetime 150 sec is about 0.9.

The hit rate depends on several factors such as the way of index propagation, the way of query propagation, and the timing of injection of altered index. Figure 6 shows time variation of the number of copies of the index of illegal file. As shown in the figure, under

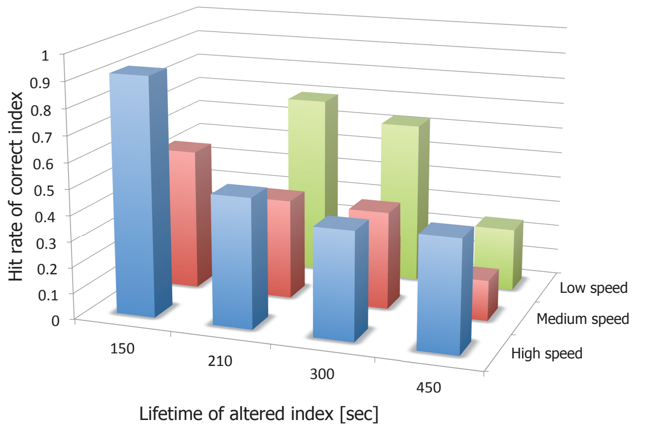


Fig. 5. Hit rate of queries requesting the correct index of illegal file.

the proposed scheme, the percentage of correct copies gradually decreases while repeating vibrations, which indicates that the hit rate strongly depends on the timing of issuing queries. Hence we need to conduct further experiments by carefully setting parameters, which is left as an important future work.

VI. Concluding Remarks

This paper proposes an index poisoning scheme for P2P file sharing systems. The proposed scheme is a combination of three different injections with different roles, and could be applied to general unstructured P2P file sharing systems as long as the direction of the flow of indices and queries can be detected. The simulation results indicate that the proposed scheme could attain almost the same performance with a simple (but expensive) scheme in which altered index with a sufficiently long lifetime is directly injected to all peers in the system.

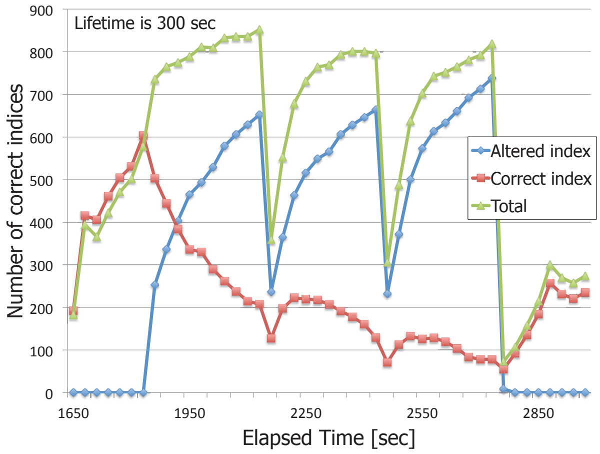
A future work is to evaluate the cost of the scheme in more detail, as well as the detailed evaluation of the performance under different settings. An application of the scheme to other unstructured P2P files sharing system such as BitTorrent and Overnet is also a crucial issue.

References

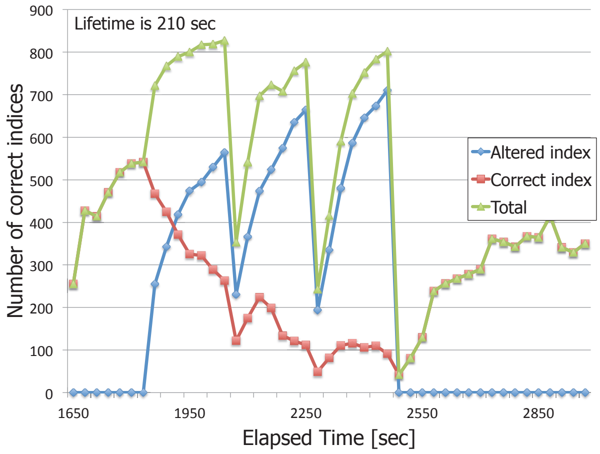
[1] E. Abdullah and S. Fujita. "Prevent Contents Leaking in P2P CDNs with Robust and Quick Detection of Colluders." *Journal of Information Processing (JIP)*, 20(2):378–385 (2012).

[2] E. Abdullah and S. Fujita. "Reputation-Based Colluder Detection Schemes for Peer-To-Peer Content Delivery Networks." *IEICE Trans. Information and Systems*, E96-D(12):2696–2703 (2013).

[3] R. J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, 2008.



(a) When lifetime is 300.



(b) When lifetime is 210.

Fig. 6. Impact of the update interval of file owner.

[4] J. Kong, W.-D. Cai, and L. Wang. "The evaluation of index poisoning in BitTorrent." In *Proc. the 2nd International Conference on Communication Software and Networks (ICCSN '10)*, 2010, pages 382–386.

[5] J. Liang, R. Kumar, Y. Xi, and K. W. Ross. "Pollution in P2P File Sharing Systems." In *Proc. the 24th IEEE International Conference on Computer Communications (INFOCOM)*, 2005, pages 1174–1185.

[6] J. Liang, N. Naoumov, and K. W, Ross. "The Index Poisoning Attack in P2P File Sharing Systems." In *Proc. the 25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006, pages 1–12.

[7] X. Lou and K. Hwang. "Collusive Piracy Prevention in P2P Content Delivery Networks." *IEEE Trans. on Computers*, 58(7):970–983 (2009).

[8] P. Putra and A. Nakao. "An Effective Index Poisoning Algorithm for Controlling Peer-to-Peer Network Applications." In *Proc. International Workshop on Modeling, Analysis, and Control of Complex Networks (Cnet '11)*, 2011, pages 17–22.

[9] S. Xie, B. Li, G. Y. Keung, and X. Zhang. "Coolstreaming: Design, Theory, and Practice." *IEEE Trans. on Multimedia*, 9(8):1661–1671, May 2007.

[10] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li. "Design and Deployment of a Hybrid CDN-P2P System for Live Video Streaming: Experiences with LiveSky." In *Proc. of the 17th ACM International Conference on Multimedia (MM'09)*, 2009, pages 25–34.

[11] M. Yoshida, S. Ohzahata, A. Nakao, and K. Kawashima. "Controlling File Distribution in Winny Network through Index Poisoning." In *Proc. International Conference on Information Networking (ICOIN 2009)*, 2009, pages 1–5.

[12] M. Yoshida, S. Ohzahata, A. Nakao, and K. Kawashima. "Controlling File Distribution in Share Network Through Content Poisoning." In *Proc. the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2010, pages 1004–1011.

[13] X. Zhang, J. Liu, B. Li, and T.-S. Peter Yum. "CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming." In *Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Vol. 3, 2005, pages 2102–2111.