An Effective Index Poisoning Algorithm for Controlling Peer-to-Peer Network Applications

Pratama Putra
Graduate School of Interdisciplinary Information Studies
The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo, Japan 113-0033
Email: qq096410@iii.u-tokyo.ac.jp

Akihiro Nakao
Interfaculty Initiative in Information Studies
The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo, Japan 113-0033
Email: nakao@iii.u-tokyo.ac.jp

***Abstract*—Copyright infringement is considered a significant issue in P2P network communications. Index and content poisoning have been proposed to control the exchange of copyrighted content. Unfortunately, however, such control methods are costly in terms of the amount of control traffic since they apply the control method to all the peers and they generate much control traffic. In general, directly applying index poisoning to a peer may indirectly poison the neighboring peers, thus, it is possible to reduce the number of peers to target for the same effect of poisoning. In this paper, we propose a method to improve index poisoning by limiting the scope of poisoning so that even when we apply poisoning to a small number of peers, we could still achieve the same effect of traffic control as we applied poisoning to all peers. In more detail, taking Winny as an example of structured P2P network, we propose, implement and evaluate an algorithm to determine the influential peers for index poisoning based on the inferred network structure. We successfully reduce the poisoning target to 27% of the total peers and achieve the same effectiveness as the conventional method poisoning 96% of the entire network.**

I. Introduction

Recently, P2P technology has become so popular that a large number of files are being exchanged by millions of users concurrently. However, due to the significant growth of P2P file sharing, even copyrighted files are also actively exchanged so copyright infringement has become a serious issue.

Index and content poisoning [3], [4], [10], [11] have been proposed to control the exchange of copyrighted content. Unfortunately, however, such traffic control methods generate a large amount of traffic and make it difficult to control the distribution of a large number of files. For example, in [10], controlling the distribution of a single file in Winny network generates 92 Kbps of transmission traffic. It is also reported that there are nearly 3 millions of copyrighted content and privacy data illegally distributed using Winny applications, hence, controlling the distribution of such content would generate traffic more than 276 Gbps. This much traffic is not acceptable since it is not negligible any more compared to the legitimate traffic in P2P networks as well as in the Internet.

In this paper, we propose a method to improve index poisoning by limiting the scope of poisoning so that even when we apply poisoning to a small number of peers, we could still achieve the same control result as that of the conventional one with the global scope of poisoning.

In a nutshell, we first identify the peers with high influence on index poisoning from the clusters of P2P networks using a clustering algorithm (more detail about the algorithm is explained in section 4). Then we apply index poisoning to the peers in the selected clusters and compare the performance with poisoning randomly selected peers of the same number and the conventional method targeting all the peers. We compare the proposed poisoning method with random and conventional ones from the following two aspects: (1) the spreading speed of the poison (poisoned file keys) and (2) coverage (the ratio of the number of peers poisoned to the total number of peers). Finally, we investigate the generated traffic of our method and compare it with the generated traffic of conventional poisoning.

II. Index Poisoning

A pure P2P network such as Winny lacks the presence of servers, thus, makes it difficult to control the distribution of files. This situation leads to the domination of copyrighted content over the legal one in the network. In addition, much traffic generated by P2P networks carries copyrighted files such as music, videos and application software. Therefore, controlling the illegal content distribution in P2P network may also contribute to reducing the traffic of P2P network in general.

A solution to controlling the file distribution in pure P2P networks is index poisoning. Index poisoning floods the network with bogus file indices. In pure P2P networks, peers exchange the information of the downloadable files through the file indices, or also known as file keys. The file key of a file contains the information related to the file such as name, size, and ID. Most importantly, it contains the location of the file owner (the IP address and the service port number), thus, enables peers to find a desired file and to contact the owner of the file for download. Polluting the network with bogus file keys hinders the peers from obtaining the information necessary for downloading the files, therefore, prevents the file from being distributed.

In Winny network, index poisoning is conducted as follows.

1) Obtain the original file key that contains the information of the target file. This can be done using the P2P

978-0-9836283-1-6 © 2011 ITC

This paper was peer reviewed by subject matter experts for publication in the Proceedings of Cnet 2011

crawlers since the file keys is packed inside the search queries [7].
2) Generate file keys with the same file ID, but replace the file owner information with the bogus one.
3) Send the modified file keys to the target peers via search queries.

III. Distributed Index Poisoning System

We develop a distributed index poisoning system that includes distributed crawlers for accurately measure P2P network topology [8] and the component that implements an algorithm for conducting effective poisoning on the Winny network using the network structure derived from the topology information obtained from the crawlers, as shown in Fig. 1.

Our distributed index poisoning system consists of the following four components.

- **Topology Crawler** for collecting the topology information (joining peers and the existing links).
- **Topology Analyzer** for implementing the influential peer selection algorithm.
- **Index Poisoner** for applying index poisoning to the peers selected by the topology analyzer.
- **File Key Crawler** for collecting the poisoning file key information for the evaluation purpose.

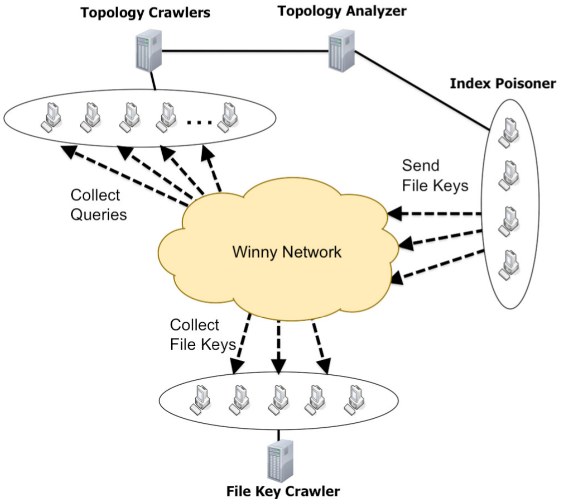

Fig. 1: Distributed Index Poisoning System.

IV. Clustering Algorithm

In this research, we use the Winny network as an example P2P network to implement our proposed effective index poisoning method. Winny is a pure P2P Network applications with nearly 100,000 peers joining per day, and is one of the most popular P2P file sharing application in Japan [6].

In this section, we introduce our method to identify highly influential cluster of peers for poisoning using clustering algorithm consisting of *level-1 clustering* and *level-2 clustering*.

A. Considering the Network Structure

In general, the peers with high degrees appear to be such effective targets, since they are connected to many other peers, thus, targeting such peers could end up poison a lot of peers indirectly. However, in structured P2P networks such as Winny, poisoning only high degree peers is not effective and we must consider the network structure for effective poisoning [9].

The network structure in Winny is mainly determined by the bandwidth declaration of joining peers [2]. The Winny protocol states that the network consists of three levels of peers. Which level a peer belongs to is determined by the bandwidth that the peer is connected at. In fact, a peer's connection bandwidth is declared by the peer itself and can be selected among 50, 120, and 1000 Kbps that correspond to the levels of downstream, middle-stream, and upstream, respectively.

Search queries are generated with higher rate from downstream to upstream. Thus, targeting the peers in the lower level of the network (50 or 120 Kbps) for index poisoning should be more effective since our index poisoning utilizes file keys carried by the search queries.

To confirm this, we conduct an experiment to compare the spread of file keys from the middle stream vs. upstream. We select a group of peers with declared bandwidth 1000 Kbps or higher and another with 120 Kbps.

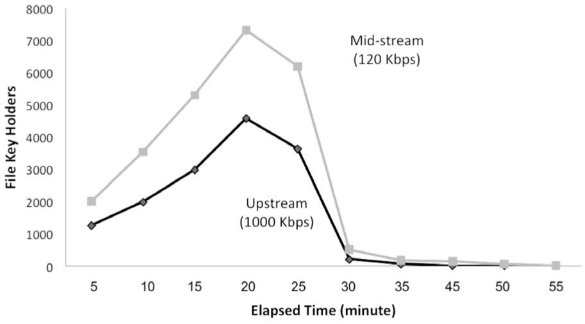

Fig. 2: Key spreading in upstream peers vs. middle stream peers

Fig. 2 shows the spread of file keys distributed to upstream and middle-stream peers. The x and y axes show the elapsed time and the number of peers holding the poison file keys (spread), respectively. From this result, we figure that it is important to poison middle stream peers mostly with bandwidth declaration of 120 Kbps since the file keys distributed from the middle stream spreads almost twice as fast and more as the file keys distributed from the upstream.

However, to obtain the declared bandwidth from a peer, it is necessary to establish connection to the peer, which is difficult since many peers (especially behind NAT) reject connection during the topology measurement [9]. Accordingly, we develop an algorithm without actually looking into the peers' bandwidth declaration where we cluster the peers with (*level-1 clustering*) using their topology characteristics that are mostly correlated with their declared bandwidths, thus, with the network structure.

After successfully inferring the network structure, we perform another clustering (*level-2 clustering*) to identify the highly influential peers for poisoning.

B. Level-1 Clustering Algorithm

In more detail of the level-1 clustering algorithm, we divide the network into 3 clusters by looking at their behavior in search queries (as shown in Fig. 3):

- ORIGIN: the peers that are not relaying queries (27% of the total peers).
- EDGE: the peers at the termination of queries (25% of the total peers).
- RELAY: the peers others than ORIGIN or EDGE (48% of the total peers).

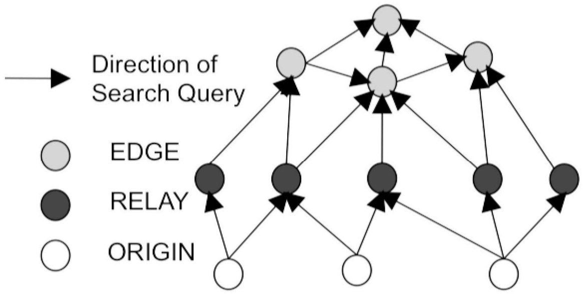

Fig. 3: The level-1 clustering algorithm

To infer these clusters, we develop a clustering algorithm using the information obtained from the search queries executed by the distributed crawlers. A search query records the information of peers in sequence, which makes it easier to classify the peers as described (ORIGIN, EDGE and RELAY). The algorithm is described in Algorithm 1:

Algorithm 1 Level-1 clustering algorithm

```
for all $Query\ Information$ do
  if $peers\ info\ count \geq 3$ then
    for all $Peer\ Information$ do
      if $Peer\ in\ First\ Record$ then
        $Classify\ as\ ORIGIN$
      else if $Peer\ in\ Last\ Record$ then
        $Classify\ as\ EDGE$
      else
        $Classify\ as\ RELAY$
      end if
    end for
  end if
end for
```

It turns out RELAY and EDGE contain the peers with 120 Kbps and 1000 Kbps bandwidth declaration, respectively. Also, ORIGIN includes many unreachable peers with unknown bandwidth, as shown in Fig. 4.

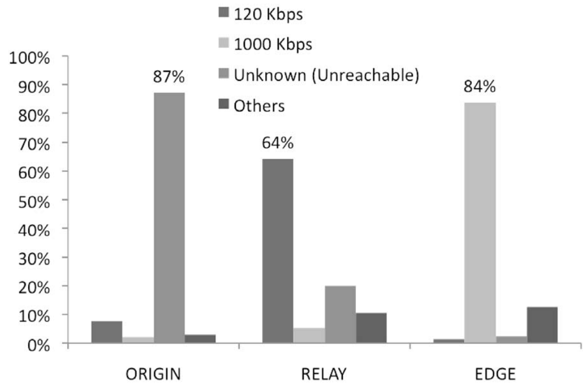

Fig. 4: The characteristics of clusters in level-1 clustering

It seems efficient to only poison RELAY peers with 120 Kbps bandwidth declaration, since we believe the low bandwidth peers are influential for indirectly poisoning neighboring peers as discussed earlier. To confirm this we conduct another experiment to evaluate the key spread of EDGE and RELAY. The experiment method is the same as the preliminary experiment with middle vs. upstream.

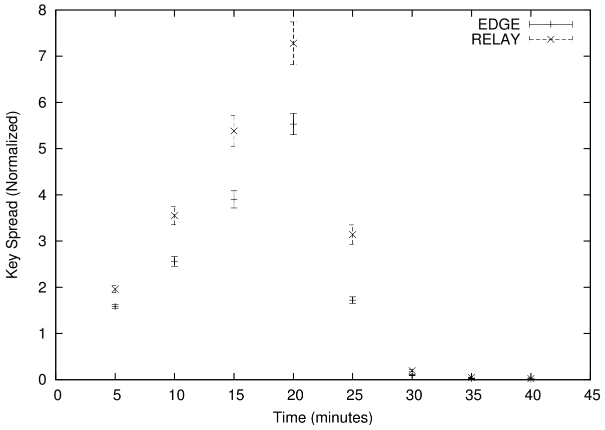

Fig. 5: Key spread of RELAY and EDGE clusters. The error bar shows the standard error from 6 times experiments.

Fig. 5 shows the comparison of key spread when applying index poisoning to RELAY and to EDGE. The x axis shows the elapsed time and the y axis shows the key spread normalized by the initial number of file keys.

From the result in Fig. 4 and Fig. 5, we confirm: (1) EDGE and RELAY mostly resemble the upstream and middle-stream since they contain many peers with bandwidth declaration of 1000 and 120 Kbps, respectively, (2) RELAY is likely to be the cluster that contains many influential peers for index poisoning since the file keys distributed to the RELAY spread more and faster.

However, since the volume of RELAY peers is nearly half of the total peers, we now consider identifying an even smaller set of influential peers among RELAY to achieve more effective poisoning.

C. Level-2 Clustering Algorithm

In level-2 clustering, we further divide RELAY cluster into the following three clusters based on their proximity of EDGE and ORIGIN cluster in the level-1 clustering (as shown in Fig. 6):

- TOP: adjacent peers to EDGE.
- BOTTOM: adjacent peers to ORIGIN.
- MIDDLE: others.

To infer these clusters, we develop an algorithm shown in Algorithm 2

Algorithm 2 Level-2 clustering algorithm

```
for all $RELAY\ peers$ do
  $Assign\ Each\ Peer\ to\ MIDDLE$
end for
for all $EDGE\ peers$ do
  for all $NeighborPeers$ do
    if $Neighbor\ Peer \in RELAY$ then
      $Assign\ Neighbor\ Peer\ to\ TOP$
      $Remove\ Neighbor\ Peer\ from\ MIDDLE$
    end if
  end for
end for
for all $ORIGIN\ peers$ do
  for all $NeighborPeers$ do
    if $Neighbor\ Peer \in RELAY$ then
      $Assign\ Neighbor\ Peer\ to\ BOTTOM$
      $Remove\ Neighbor\ Peer\ from\ MIDDLE$
    end if
  end for
end for
```

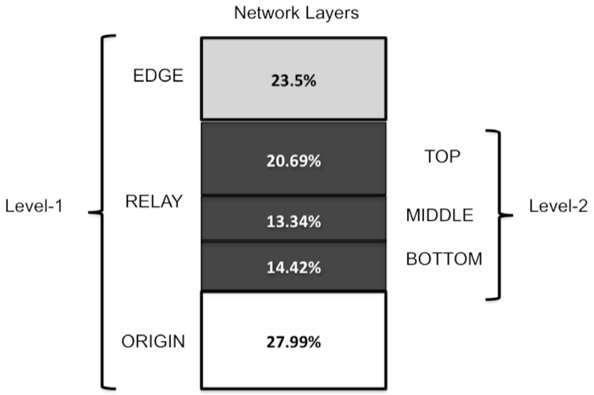

Fig. 6: Level-1 and level-2 clustering

We evaluate the performance of the three clusters in level-2 clustering by distributing different file keys to the peers in each cluster in similar ways to that in performance evaluation in the level-1 clustering.

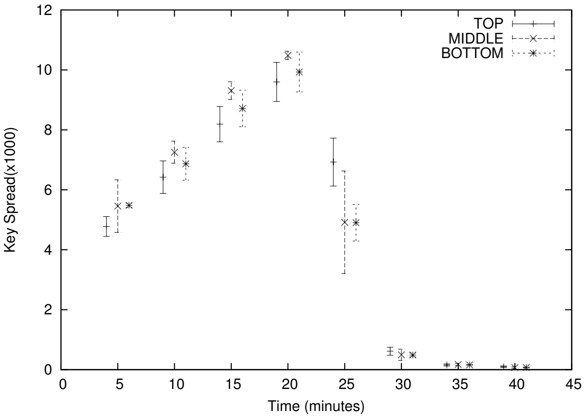

Fig. 7: TOP vs. MIDDLE vs. BOTTOM. The error bar shows the standard error from 4 times experiments.

Fig. 7 shows the comparison of key spread in the three clusters. The x and y axes both show elapsed time and key spread, respectively. As shown in Fig. 7, we conclude that the MIDDLE is the most effective cluster for index poisoning for the following three reasons: (1) it contains many peers with 120 Kbps bandwidth declaration, (2) it occupies the smallest portion of peers (MIDDLE 13%) and (3) we confirm from the result in Fig. 7 MIDDLE shows the best performance since the file key spread more and faster.

V. Evaluation

A. Evaluation Setting

We evaluate four poisoning methods by conducting index poisoning experiments with the same methodology as the one we conduct in the previous section, where we directly poison a number of peers selected using the methods and observe the spread of keys and coverage poisoning area of the poisoning method (indirect poisoning effect). We also investigate the traffic generated to conduct poisoning using those methods. The four methods we evaluate are:

- MIDDLE poisoning: directly poison 13% of peers
- MIDDLE+BOTTOM poisoning: directly poison 27% of peers (MIDDLE 13% and BOTTOM 14%)
- Random selection poisoning: to be compared with MIDDLE poisoning, directly poison the same number of peers as MIDDLE (13%)
- All peer poisoning (Conventional poisoning): directly poison 100% of peers

From the above four methods, our proposed methods are the first two, MIDDLE and MIDDLE+BOTTOM poisoning.

B. Evaluation Result

First, we evaluate the spread of index poisoning file keys from the four poisoning methods. The key spread is defined as the number of peers holding the index poisoning file keys during the observation. We set the key lifetime to be 20 minutes and observe the key spread every 5 minutes.

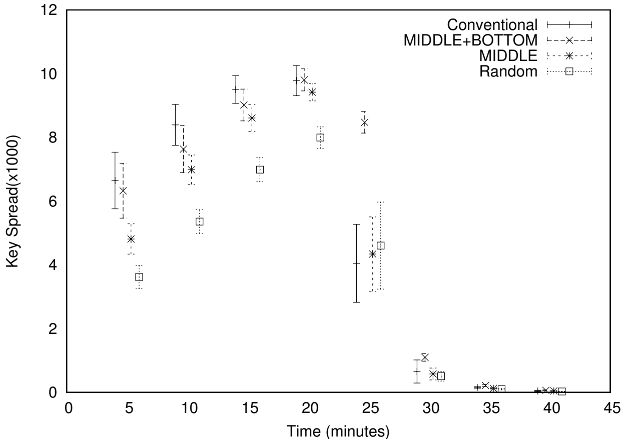

Fig. 8: Key spread evaluation. The error bar shows standard error from 5 times experiments

Fig. 8 shows the result of key spread evaluation. The x axis shows the elapsed time and the y axis shows the key spread. From the figure, we figure that poisoning MIDDLE cluster is better than poisoning the same number of peers selected randomly since the file key in the MIDDLE cluster poisoning spreads more and faster. Unfortunately, we still can not obtain the same poisoning effect as the conventional method targeting all the peers when we poison only the MIDDLE cluster.

However, combining the MIDDLE and BOTTOM cluster can improve the key spread and eventually achieves the same index poisoning effect as the conventional method. This is indicated by the close maximum key spread speed of MIDDLE+BOTTOM of the conventional.

Next, we observe the coverage area of the poisoning effect that is defined as the number of peers holding the file keys over the number of all the peers discovered during the observation. We plot the coverage as the CDF of the key spread every 5 minutes observation.

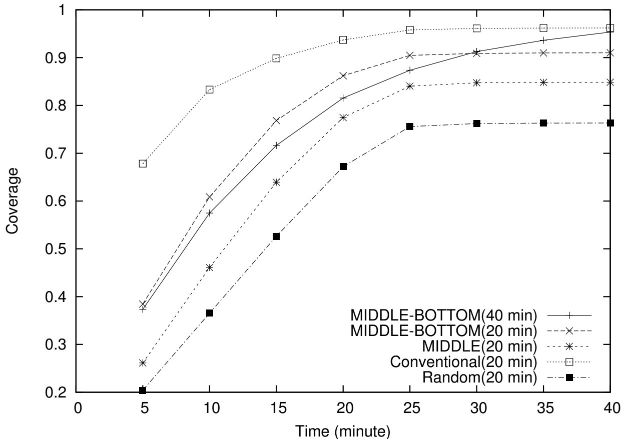

Fig. 9: Coverage evaluation.

Fig. 9 shows the coverage evaluation result where x axis shows the elapsed time and y axis shows the coverage. Fig. 9 indicates that when we set the lifetime of file key to 20 minute, the conventional method exhibits the highest coverage area of index poisoning, which is 96%. In this condition, our proposed methods can achieve high coverage such as 86% for the MIDDLE poisoning and 90% for the MIDDLE+BOTTOM poisoning, but both methods still can not achieve the same poisoning effect as the conventional method.

However, increasing the lifetime of file keys gives more duration for indirect index poisoning effect to take place in the network and the MIDDLE+BOTTOM poisoning can achieve the same coverage with the conventional method (96%).

Finally, to verify that reducing index poisoning target can reduce the generated traffic, we conduct separated experiments where we measure the outgoing traffic when we distribute index poisoning file keys to a set numbers of peers for controlling distribution of a single file.

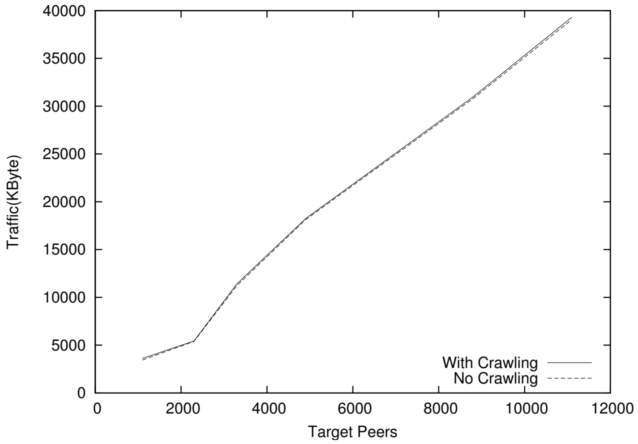

Fig. 10: Traffic evaluation.

Fig. 10 shows the number of target peers in x axis and the generated traffic in y axis. From the result, we can see that the generated traffic increases as the number of target peers. We observe the traffic with and without the topology crawling traffic. It turns out that the crawling traffic is much smaller than the poisoning traffic, so both lines in Fig. 10 are almost overlapped.

Here we can estimate that the conventional method generates more than 44MB of traffic to control the distribution of a single file, since it usually targets 12—14K peers directly. With our proposed method, we only have to target directly 2.4K peers for MIDDLE poisoning and 4.4K peers for MIDDLE+BOTTOM, thus we can reduce the traffic to 1/3 (7–15MB). From this result, we confirm that our proposed method can lift the limitation in the number of controllable files three times larger with the same amount of traffic as the conventional method.

VI. Related Work

Index poisoning method has been proposed for pure and hybrid P2P network applications in [3], [4], [10], [11]. However, none of these methods report the use of topology information to select influential peers for the index poisoning target, but instead, they apply index poisoning to all the peers in the network, thus generates high control traffic. However, they report successful index poisoning for controlling file distribution in P2P network.

Beside index poisoning, there are other methods such as content poisoning [5], [11] and pollution method [1] for controlling illegal content distribution in P2P network. Content poisoning differs from index poisoning in that instead of targeting the file indexes, it corrupts the fragments of a file. Pollution corrupts complete files, hence makes them unusable and distribute the corrupted files to the network. Some of these methods are suitable for specific P2P network, for example, [11] reports that content poisoning is more effective for Share networks than index poisoning, while content poisoning is not so effective in Winny networks, because the Winny protocol uses different links from search links to download a file and does not divides chunks to many users.

VII. Conclusion

In this paper, we present the clustering algorithm which consists of level-1 clustering that can define the network clusters (EDGE and RELAY) which nearly resembles the Winny structure (middle and upstream) without collecting peers bandwidth information and the level-2 clustering that can determine the most effective clusters (MIDDLE and BOTTOM) for conducting index poisoning. We also propose the MIDDLE+BOTTOM poisoning that can reduce the generated control traffic to 27% compared to the conventional method poisoning all the peers in the network while achieving the same coverage area of the poisoning effect which is 96%.

In conclusion, the proposed method can lift the limitation in the number of controllable files in conventional index poisoning since it minimizes the control traffic but still achieves the same effectiveness with the existing work.

For future work, we plan to improve the proposed method by reducing the index poisoning target to an even smaller number of peer to increase the effectiveness to enable controlling distribution of more files in the network. We also plan to implement our proposed method to other control methods and to P2P applications.

Acknowledgment

This work is partly supported by KAKENHI 21300020 Grant-in-Aid for Scientific Research (B).

References

[1] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena. The Pollution Attack in P2P Live Video Streaming: Measurement Results and Defenses. *Proc. P2P-TV 2007*, pages 323–328, 2007.

[2] Isamu Kaneko. *The Technology of Winny*. ASCII, 2005.

[3] J. Kong, W. Cai, and L. Wang. The Evaluation of Index Poisoning in BItTorrent. *Proc. IEEE ICCSN*, pages 382–386, 2010.

[4] J. Liang, N. Naoumov, and K. W. Ross. The Index Poisoning Attack in P2P File Sharing Systems. *IN INFOCOM*, pages 1–12, 2006.

[5] X. Lou and K. Hwang. Proactive Content Poisoning To Prevent Collusive Piracy in P2P File Sharing. *IEEE Transactions on Computers TC 2008*, 2008.

[6] Terada Masato, Ukai Yuji, Kanai Ryoji, Hatada Mitsuhiro, Matsuki Takahiro, and Miyagawa Yuichi. P2P Network Observation Using Crawling Method. *IPSJ SIG Notes*, pages 51–56, 2007.

[7] P. Putra, M. Yoshida, S. Ohzahata, A. Nakao, and K. Kawashima. Winny Network Topology Measurement using File Search Queries. *IEICE Tech. Rep.*, 109(3):25–30, April 2009.

[8] Pratama Putra and Akihiro Nakao. Measuring P2P network topology through geo-location-aware distributed crawlers. *8th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT)*, pages 1–6, 2010.

[9] Pratama Putra and Akihiro Nakao. Topology-aware Traffic Control System for P2P Networks. *IEICE Tech. Rep.*, 110(240):87–92, 2010.

[10] M. Yoshida, S. Ohzahata, A. Nakao, and K. Kawashima. Controlling File Distribution in Winny Network through Index Poisoning. *Proceedings of the 23rd International Confer-ence on Information Networking*, pages 210–214, 2009.

[11] Masahiro Yoshida, Satoshi Ohzahata, Akihiro Nakao, and Konosuke Kawashima. Controlling File Distribution in Share Network Through Content Poisoning. *Proc. of IEEE AINA2010*, 2010.