IEICE TRANS. INF. & SYST., VOL.E96–D, NO.12 DECEMBER 2013

PAPER Special Section on Parallel and Distributed Computing and Networking

Reputation-Based Colluder Detection Schemes for Peer-to-Peer Content Delivery Networks**

Ervianto ABDULLAH[†*], *Nonmember and* **Satoshi FUJITA**[†a)], *Member*

SUMMARY Recently Peer-to-Peer Content Delivery Networks (P2P CDNs) have attracted considerable attention as a cost-effective way to disseminate digital contents to paid users in a scalable and dependable manner. However, due to its peer-to-peer nature, it faces threat from "colluders" who paid for the contents but illegally share them with unauthorized peers. This means that the detection of colluders is a crucial task for P2P CDNs to preserve the right of contents holders and paid users. In this paper, we propose two colluder detection schemes for P2P CDNs. The first scheme is based on the reputation collected from all peers participating in the network and the second scheme improves the quality of colluder identification by using a technique which is well known in the field of system level diagnosis. The performance of the schemes is evaluated by simulation. The simulation results indicate that even when 10% of authorized peers are colluders, our schemes identify all colluders without causing misidentifications.

key words: *Peer-to-Peer, content delivery network, colluder detection, reputation, PMC model*

1. Introduction

Recently Peer-to-Peer Content Delivery Networks (P2P CDNs) have attracted considerable attention as a way to deliver large digital contents to many users in a cost-effective and dependable manner [4], [5], [18]. Contents delivery in a P2P CDN is typically invoked by the owner of a media file by pushing a copy of the file to the P2P overlay, and once a copy is being available in the network, it will be delivered to the authorized recipients by repeating local communications among nearby peers. In the literature, a number of contents delivery systems have been proposed and the effectiveness of such systems is quantitatively evaluated by simulation. However, although it would be certainly effective compared with traditional Client/Server systems, due to its "peer-to-peer" nature, P2P CDN face several critical issues if we wish to deploy it commercially [14]. In particular, a quick identification and exclusion of **colluders** is a crucial issue in (commercial) P2P CDNs, where colluder means a peer which paid for the contents but tries to illegally share the contents with other unauthorized peers.

Meanwhile, in open distributed systems such as P2P CDNs and P2P file sharing systems, the *reputation of peers* plays an important role in deciding an appropriate action of individual peers. In fact, in order to avoid the risk of conducting suspicious transactions, each peer should prefer to the interaction with good-reputation peers rather than the interaction with bad-reputation peers. Thus far, a number of P2P reputation systems have been proposed in the literature, which includes EigenTrust [7], PeerTrust [17], PowerTrust [21], and others. Thus, it seems to be a reasonable approach to apply such P2P reputation systems to detect colluders in P2P CDNs, as we could compare trustworthy peers to good-reputation peers and colluders to bad-reputation peers.

Unfortunately however, we can not directly apply those techniques to the collusion detection in P2P CDNs since conventional P2P reputation systems are aimed to merely protect each peer from being interacted with malicious peers [6], [12]. In contrast, the objective of colluder detection in P2P CDNs is to protect the right of content owners (i.e., copyright holders) by identifying malicious peers and by proactively excluding them. Thus, in order to apply P2P reputation systems to the colluder detection problem, we need to significantly increase the quality of colluder detection in such a way that *it identifies almost all colluders without misidentifying non-colluder peers*. In our previous paper [1], we introduced the notion of reputation to improve the quality of the decoy-based colluder detection scheme proposed by Lou and Hwang [10]. However, our previous scheme merely qualifies the peers to be selected as decoy and the qualification of other peers was remained open in our previous paper.

In this paper, we propose two colluder detection schemes. Our main concern in designing a high quality colluder detection scheme is as follows:

1. We need to detect as much colluders as possible in an early stage of the colluder detection, as undetected colluder keeps leaking contents illegally.
2. We need to avoid wrong detection of trustworthy peers, as they pay for the contents and act legally.
3. We have to keep the overhead as low as possible.

The first scheme tries to identify a set of colluders using the notion of reputations, as in our previous scheme. More concretely, we introduce two types of reports concerned with the trustworthy of the target peer, called normal report and decoy report, and update scores representing the trustworthiness of peers by successively receiving such re-

Manuscript received January 7, 2013.
Manuscript revised March 27, 2013.

[†]The authors are with the Graduate School of Engineering, Hiroshima University, Higashihiroshima-shi, 739–8527 Japan.
[*]Presently, with NTT Data Corporation.

**Earlier version of this paper was presented at "Colluder Detection in Commercial P2P CDNs Using Reputation Information," by Ervianto Abdullah and Satoshi Fujita, in Proc. Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, November 2012.

a) E-mail: fujita@se.hiroshima-u.ac.jp

DOI: 10.1587/transinf.E96.D.2696

ports (the reader should note that in the proposed schemes, evaluation of target peers is conducted by each peer although those reports are accumulated to a centralized manager). The second scheme adopts a model originally proposed for the system level diagnosis of autonomous systems to increase the accuracy of colluder identifications. The possibility of misidentifications can be bounded very small by using the second scheme, although there is a trade-off between the accuracy and the detection speed since the accuracy of colluder detection significantly degrades if the number of collected reports becomes small. The performance of the proposed schemes is evaluated by simulation. The simulation results indicate that the first scheme successfully detects almost all colluders if 10% of peers are colluders, and can keep the number of misidentifications at zero even if 30% of peers are colluders. On the other hand, the second scheme detects all colluders without making misidentification if 10% of peers are colluders, although it makes few misidentifications if 30% of peers are colluders.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 describes a model of P2P CDNs as well as a model of colluders. Sections 4 and 5 describe the details of the proposed schemes. Section 6 describes the simulation results. Finally, Sect. 7 concludes the paper with several topics for future work.

2. Related Works

There are several proposals concerned with P2P CDNs. In contrast to conventional CDNs based on the Client/Server model, the content delivery in P2P CDNs is realized by the mutual exchange of shared files among participating client peers. Although such a P2P technology could improve the system performance as well as the availability and the scalability [2], there are several issues to be addressed if we wish to deploy it as a commercial system [14]. One of such crucial issues is an illegal sharing of the contents among peers, which results in serious disadvantages such as copyright violations.

A typical approach to realize a secure content delivery in P2P environment is to use Digital Right Management (DRM) technology [8], [15], [19], [20]. DRM controls the usage and the redistribution of digital contents using several key technologies such as public-key cryptography. While the use of DRM certainly realizes a content protection, it still faces several limitations such as the platform compatibility and the trade off between privacy and anonymity of the users [9], [11].

Another approach to fight copyright violations in P2P CDNs is to "detect" an illegal sharing of the contents during the distribution process, which is expected to be more effective and more user-friendly than DRM-based approaches. The reader should note that in constructing commercial content delivery systems, we can use DRM with reputation-based schemes in a combined manner to keep the security of the resulting systems high. To date, there are only few colluder detection schemes designed for P2P CDNs. Lou and Hwang proposed a randomized scheme with a proactive contents poisoning method [10]. Their scheme conducts a colluder detection based on a set of reports collected from client peers similar to existing P2P reputation systems, although it assumes that all colluders behave in a lenient way such that they respond to any request received from the other peers while such a situation rarely happens in the real world.

Sherman *et al.* took a different approach called **trusted auditing** to detect colluders in trusted P2P CDNs [16]. In their approach, contents owner operates Trusted Auditors (TAs) to help finding colluders in the network. Such a hybrid approach incurs contents owners an additional cost to operate TAs, which will become higher as the number of client peers increases. In addition, although it could recruit TAs from the set of client peers in order to overcome such a scalability issue, it is generally difficult to find fully trusted peer to be recruited as a TA. In this paper, we propose a colluder detection scheme for common P2P CDN platforms similar to [16]. Our scheme is based on a P2P reputation system which leaves the task of colluder detection to the clients in the network, and instead of taking a hybrid approach as in [16], we adopt a centralized server called Management Server (MS) to control the overall reputation management including the aggregation of reputation reports. More specifically, in our scheme, the role of TAs is divided into two parts, i.e., the check of the status of target peers and the calculation of suspicious peers from the outcome of such checks, and the first role is distributed to all peers participating in the network. The reader should note that the load of the second task is much lower than the load of the first task.

3. Model

This section describes a model of colluders in (commercial) P2P CDNs. See Fig. 1 for illustration. To purchase a digital content, a user contacts a centralized content provider's portal (Step 1) and after completing the payment, she will receive a "token" corresponding to the purchased contents (Step 2) which is used to authorize her to download the purchased contents. Then the peer joins the P2P network to share the content to other authorized peers (Steps 3 and 4), while a colluder may illegally share the content to unauthorized peers (Step 5). During the file transfer, peers authenticate one another to make sure that they share the file only to authorized peers, which is verified by the token obtained on purchasing[†].

If all peers in the P2P network are trustworthy, it is not difficult to allow authorized peers to successfully receive requested files and to reject any download request received from unauthorized peers. Unfortunately however, in actual

[†]In the literature, several authentication methods during the download process have been proposed [10], [16], but we do not consider such methods in detail because it is beyond the scope of our work. In this paper, we will simply assume that such an authentication correctly works and each peer could easily verify the authorization of peers to any file.

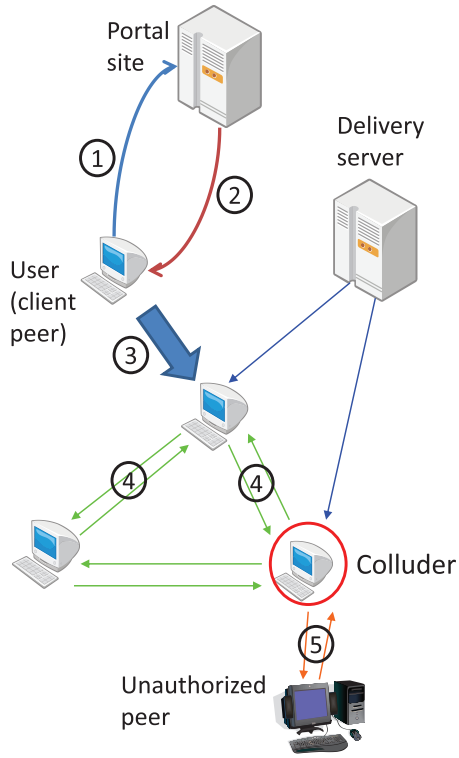

Fig. 1 Procedure of content purchasing in P2P CDNs with a colluder.

P2Ps, there may exist several "dishonest" peers called **colluders** which illegally share purchased contents to unauthorized peers. Since the existence of colluders significantly loses the benefit of the owner of paid contents, it is strongly required to identify all colluders and punish them if necessary.

Such a leaking of contents can be easily stopped by removing the access right from the colluder if there is a mechanism to identify colluders. Thus in the following, we assume that there is no such colluder identification mechanisms. In our model, each colluder is assumed to follow Sherman's observation such that each colluder may illegally share purchased contents to other peers because *they expect those peers to share other paid contents to them in return* [16]. More concretely, each peer is assumed to conduct the following steps [16] (see Fig. 2 for illustration):

1. Each colluder tries to find other colluder in the network to share with, by probing some of its neighbours (arrow 1 in Fig. 2).
2. If some peer responds to the probe (arrow 2), it recognizes them as **fellow colluders**, and asks them to form a cluster to illegally share contents among members of the cluster (arrow 3).
3. To avoid being detected easily, each colluder chooses a finite strategy called growth factor, which reflects the minimum cluster size to which it aims to belong at the end of the download session.
4. The selected value of the growth factor, which is referred to as GF hereafter, affects the behaviour of the colluder in the following manner: 1) it repeats probing and keeps responding to any incoming probe until it discovers at least $GF - 1$ other colluders, and 2) after finding $GF - 1$ fellow colluders, it stops probing, and in the succeeding steps, it refuses any probe received from other peers.

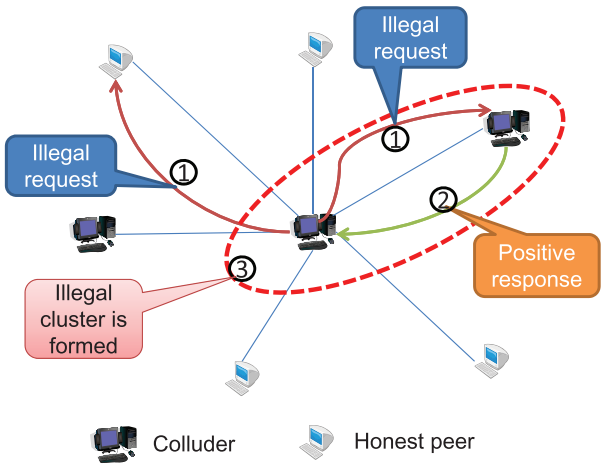

Fig. 2 Model of colluders.

In this paper, we do not discuss the way of determining an appropriate value of parameter GF because it is out of scope of our proposal.

4. First Scheme

4.1 Outline

In the first scheme, the reputation of peer i is represented in the form of a **reputation score** r_i ($\in [0, 1]$), which is calculated by aggregating two different types of reports received from peers in the system, i.e., normal report and decoy report. The outline of the scheme is described as follows.

- Each peer periodically reports the behavior of its neighbors in the P2P overlay in the form of normal report to the centralized server called the Management Server (MS), where each report is weighted with the reputation score of the reporter. Then any peer which tried to form a cluster with other colluders is penalized by decreasing the reputation score.
- Each peer probabilistically acts as a decoy and asks its neighbors to form an illegal cluster. Neighbors who agreed to form an illegal cluster will be penalized by decreasing the reputation score, where such an action is notified to the MS by the decoy in the form of decoy report.
- A peer whose reputation score becomes lower than a predetermined threshold is regarded as a colluder, and is removed its authorization.

More concretely, the reputation score of each peer is calculated by using another score c_i called **collision score** in the following manner:

$$r_i := 1 - \frac{c_i}{\gamma}, \quad (1)$$

where γ (> 0) is a constant called collusion threshold and the way of updating c_i with normal and decoy reports will be described in the succeeding subsections. The reader should note that all scores, along with peers' authorization are maintained by the MS.

4.2 Normal Report and Decoy Report

In P2P CDNs, each peer periodically updates the set of neighbors in the P2P overlay. For example, the choke algorithm used in BitTorrent protocol updates the set of neighbors every 20 seconds by default [3]. A session for a peer is the time period which starts when the peer connects to its current neighbors and ends when it disconnects the link with some of its neighbors. **Normal report** concerned with each neighbor is issued at the end of each session and specifies whether or not the neighbor has issued a request to form an illegal cluster during the session, i.e., it takes a binary value.

In the proposed scheme, MS periodically hires several peers as decoys to mimic the behavior of colluders, i.e., each decoy broadcasts a request to form an illegal cluster to its neighbors. If there is a neighbor which returns a positive response to the request, the decoy identifies the neighbor as a malicious peer and notifies the fact to MS with a **decoy report**, which takes a binary value as in normal reports. Decoy peers are randomly selected from the set of high reputation peers, where peer i is regarded as a high reputation peer if $r_i > \lambda$ for a predetermined threshold $\lambda \in (0, 1)$. Note that the set of high reputation peers dynamically changes since the reputation score r_i will be updated along with the contents of aggregated normal and decoy reports.

4.3 Procedures

All reports issued by the peers (including decoys) are aggregated to MS. Consider the following two binary variables q_{ij} and p_{ij} concerned with the behavior of peer i reported by peer j ($\neq i$):

- Variable q_{ij} indicates whether i asked j to form an illegal cluster in a session. Namely $q_{ij} = 1$ if and only if i issued such a request and i is not a decoy.
- Variable p_{ij} indicates whether i responded to a request issued by decoy j. Namely $p_{ij} = 1$ if and only if i returns a positive reply to a request issued by decoy j.

Intuitively, $\sum_j (p_{ij} + q_{ij})$ represents the total number of votes indicating i is a malicious peer, and i is likely to be a colluder if the value becomes large. However, a colluder might issue false reports to protect fellow colluders (including itself) from being detected by MS. If colluders *always* falsely report their neighbors' behavior, i.e., if they report silent neighbors as a peer asking to form an illegal cluster and do not report any malicious behavior of its neighboring colluders, such false reports can significantly affect the entire reputation score. To reduce the impact of such false reports, we normalize the reported values received from peer j by the total number of reports issued by j. More concretely, instead of using raw value "$p_{ij} + q_{ij}$," we use a normalized value n_{ij} defined as follows:

$$n_{ij} := \frac{p_{ij} + q_{ij}}{\sum_k (p_{kj} + q_{kj})}. \quad (2)$$

Note that such a normalization ensures that the sum of values reported by j is between 0 and 1, and limits the possible damage due to false reports received from malicious colluders.

Based on such values received during a session, MS updates the collision score c_i of i introduced in the previous session as follows:

$$c_i := \min\left\{ c_i + \sum_j (r_j \times n_{ij}), \gamma \right\},$$

where r_i is the reputation score updated using Eq. (1) and γ is the collision threshold. Note that each normalized report is weighed with reporter's reputation score, to ensure that the report received from malicious peers is less effective.

5. Second Scheme

5.1 Outline

The first scheme implicitly assumes that the report issued by a colluder is not correct merely in a probabilistic sense and it is unlikely that a colluder tricks a trustworthy peer into a low reputation in collusion with the other colluders. Our second scheme is designed to be resistant to such a "collusion of colluders."

The key idea is to adopt a model used for the system level diagnosis which was proposed by Preparata, Metze, and Chien in 1967 [13] (it is called the PMC model in the literature). By applying the PMC model to the reputation management problem, we could precisely identify a set of colluders as long as: 1) any non-colluder peer can correctly check whether a given peer is colluder or not, and 2) the number of colluders does not exceed the majority of the peers. In the following, after briefly describing the PMC model, we describe the details of the proposed colluder detection scheme based on the PMC model.

5.2 PMC Model

Let us consider a collection of peers. The PMC model assumes that the state of each peer is either "healthy" or "faulty," where the healthy state is represented by value 0 and the faulty state is represented by value 1. For any peers i and j, peer i can test the state of peer j ($\neq i$) and the test result is represented by a binary variable x_{ij}, where

1. if the tester i is healthy, then x_{ij} correctly reflects the state of j, i.e., $x_{ij} = 1$ if and only if j is faulty, and
2. if the tester i is faulty, then x_{ij} takes an arbitrary value.

In other words, it assumes that the test result of healthy peer

is always correct while the test result of faulty peer is not reliable. In our setting, we could make a natural correspondence between faulty peers and colluders, i.e., we can simply assume that trustworthy peers are healthy and colluders are faulty.

5.3 Procedures

In actual P2P reputation systems, we can not guarantee that every trustworthy peer correctly identifies the set of colluders as in the PMC model, since a colluder might not always send a message to form an illegal cluster to the neighbours, nor always return a positive response to a probe message given by a decoy (in contrast, it is always true that if peer i reports that j is a colluder, then at least one of i and j is a colluder, i.e., either peer i correctly recognizes that j is a colluder or colluder i tells a lie). In order to overcome such an issue, in the second scheme, we combine our first scheme with the PMC model. More concretely, our second scheme works as a "post-process" of the first scheme and it is invoked when the number of reports concerned with each peer reaches a predetermined threshold α (> 0).

Let z_i be the number of reports indicating that peer i is a colluder. If $z_i \geq \alpha$, we judge that i is a colluder and neglect any report issued by i. If $z_i = 0$, we judge that i is trustworthy and believe all reports issued by i. Let V be the set of peers and let $N = |V|$. Thus in the following, without loss of generality, we assume 1) $0 < z_i < \alpha$ for any $i \in V$, and 2) the number of colluders is at most $N/2 - 1$ (the reader should note that the second condition is necessary to correctly identify a set of colluders). A procedure to recognize a set of colluders in V conducted by the MS proceeds as follows. At first, it identifies a maximal set of peers U ($\subset V$) satisfying the following property:

- For any peers i and j in U, there is a sequence of reports which certifies the trustworthy of j by i in a transitive manner, where the word "transitive" means that if i recognizes j is trustworthy and j recognizes k is trustworthy, then i transitively recognizes k is trustworthy (note that such a sequence of reports must exist in both directions, i.e., from i to j and from j to i).
- U does not contain two peers i and j such that i issues a report indicating that j is a colluder.

It is known in the literature that if the number of colluders is less than $N/2$, then there is a subset U such that the cardinality of U is greater than $N/2$. Thus, after identifying maximal subset U (with size exceeding $N/2$), we judge that all peers in U are trustworthy and then all peers not in U are colluders.

Of course, the performance and the accuracy of the resulting scheme is sensitive to the selection of parameter α. In fact, a larger α results in a slower detection speed, since it must collect at least $\alpha \times N$ reports before starting the colluder identification step. A slower detection speed increases the probability of colluders to successfully form an illegal cluster, and under our model of colluders, once such a cluster is formed, it will never be detected by the other peers including MS. On the other hand, a smaller α results in a lower accuracy, since it would easily misidentify colluders as an trustworthy peer (recall that it judges peer i is trustworthy if $z_i = 0$). In addition, such a misidentification increases the risk such that a trustworthy peer is misidentified as a colluder, since the scheme is designed to believe all reports issued by a peer which has been judged to be trustworthy. As trustworthy peer pays for the contents and acts legally, we should keep the possibility of such a misidentification as small as possible. The impact of parameter α to the performance of the scheme will be experimentally evaluated in the next section.

6. Simulation

In this section, we evaluate the performance of the proposed schemes with respect to the accuracy, detection speed, and the overhead.

6.1 Setup

We consider a P2P CDN consisting of 1000 homogeneous peers. For each peer, the number of neighbors is maintained to be 10 to 20, and the link connecting to three selected neighbors is used to upload chunks of the shared contents. Such selected neighbors are said to be unchoked and the remaining neighbors are said to be choked. The selection of unchoked peers is conducted according to the Tit-for-Tat strategy adopted in the BitTorrent protocol, and links connecting to choked neighbors are used only for the exchange of the chunk availability information. For every minute, each peer randomly updates the set of choked neighbors.

The percentage of colluders, which is referred to as the **collusion rate**, is varied from 10% to 40%, where 10% indicates that 100 peers among 1000 peers are colluders. Parameter GF is fixed to five. For every minute, each colluder issues a probe message to its two to five random neighbors until it forms an illegal cluster of size (at least) five. Any report issued by a colluder is incorrect. That is, it maliciously reports its silent neighbors as colluders and does not report any malicious behavior of its neighboring colluders. The reader should note that this corresponds to the most difficult situation for the colluder detection scheme based on the reputation information. For every two minutes, MS hires ten random peers as decoys (i.e., 1% of the population are selected as decoys), where the term of service as a decoy is two minutes. If a colluder is accidentally chosen as a decoy, it sends wrong reports concerned with its neighbors.

Parameters specific to the first scheme are fixed as follows. The initial score of each peer is given as $r_i := 1$ and $c_i := 0$ for each i. The collision threshold γ used in Eq. (1) is fixed to 5 and threshold λ which is used to recognize a peer as a high reputation peer, is fixed to 0.7.

To determine an appropriate value of parameter α, which is a specific parameter to the second scheme, we conducted a preliminary experiment to evaluate the impact of α

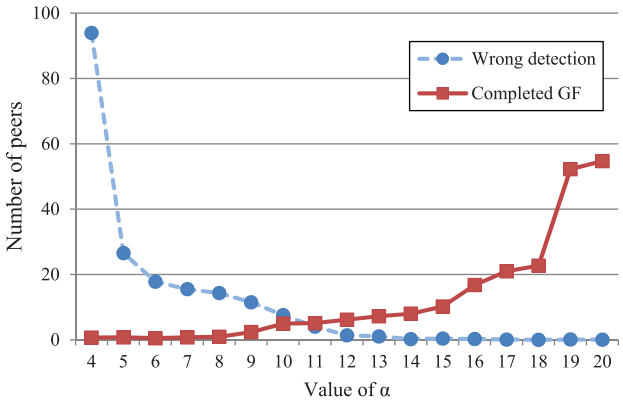

Fig. 3 Impact of α to the performance of the second scheme.

Table 1 The performance of the first scheme.

Collusion rate [%]	Detected colluders [%]	Wrong detection	Completed GF
10	99.7	0	0.4
20	94.6	0	11.7
30	77.8	0	70.4
40	57.9	0	174

to the detection speed and the accuracy by fixing the collusion rate to 30%. Figure 3 summarizes the results. In the figure, "wrong detection" indicates the number of misidentifications of trustworthy peers as a colluder and "Complete GF" means the number of colluders which successfully formed an illegal cluster of size five. For each metric, a lower value would be better. As shown in the figure, there is a trade-off between the detection speed and the accuracy; i.e., although we could identify all colluders for $\alpha \leq 6$, it significantly increases the number of wrong detections (e.g., 18 trustworthy peers are misidentified as a colluder at $\alpha = 6$), and on the other hand, although we could avoid such a misidentification for $\alpha \geq 14$, it gradually increases the number of illegal clusters which could not be detected by the scheme. In the succeeding simulations, we focus on the number of misidentifications as the primary metric, and set the value of α to 14.

6.2 Result

At first, we evaluated the speed of colluder detection. Table 1 summarizes the results for the first scheme, where each value is an average over one hundred runs. As shown in the table, when the collusion rate is 10%, it detects almost all colluders, while it allows few colluders to form an illegal cluster. The number of wrong detections is kept to be zero. The performance of the scheme becomes worse as the collusion rate increases, and when the collusion rate is 30%, it could detect only 78% of colluders, namely it misses about 70 colluders (in fact, it allowed 70 colluders to successfully form an illegal cluster). The reader should note that the number of undetected peers is smaller than the number of peers which successfully form an illegal cluster, which is because few peers are identified as a colluder (immediately) after forming a cluster.

Table 2 The performance of the second scheme.

Collusion rate [%]	Detected colluders [%]	Wrong detection	Completed GF
10	100	0	0
20	99.9	0	0.44
30	99.4	0.22	7.31
40	63.3	1.85	174.3

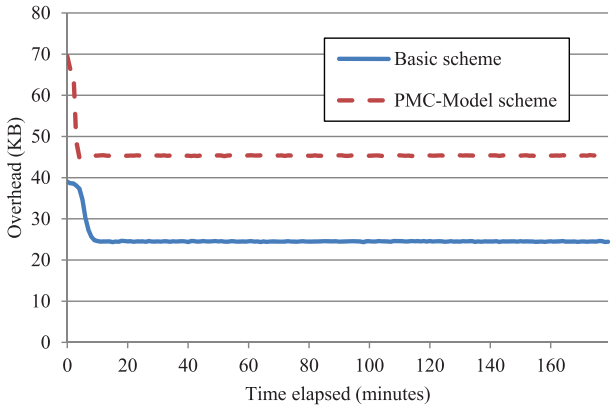

Fig. 4 Overhead of the proposed schemes.

The result for the second scheme is summarized in Table 2, which is an average over 50 runs. It is remarkable that *the scheme detects all colluders without making any misidentification if the colluder rate is 10%*. Although the number of missed colluders slightly increases as the colluder rate increases, it is still very small compared with the first scheme. A weak point of the second scheme is that it causes very small number of misidentifications if the colluder rate is high, which is because the value of parameter α used in the experiment was too small to detect all colluders under such a high collusion rate.

A small communication overhead is another key concern in designing colluder detection schemes applicable to P2P CDNs. Figure 4 shows the time transition of the communication overhead due to normal and decoy reports per minute incurred by the proposed schemes assuming that the overhead of each message is 10 KB and the collusion rate is 30%. Although the second scheme causes larger overhead than the first scheme, the amount of the communication overhead is kept small (e.g., few MB in total) compared with the total bandwidth necessary for the content delivery. The reason of the badness of the second scheme is explained as follows. In both schemes, each peer issues a normal report every session, while such an issue can be skipped in the first scheme if all neighbors are recognized as trustworthy in the last session (recall that the first scheme merely counts the number of reports indicating the maliciousness of the target peer).

The higher communication overhead in an earlier stage of the simulation is due to the following reasons: 1) the number of normal reports issued by honest peers gradually decreases since we are assuming that the issue of normal

reports is skipped if all neighbors are recognized to be trustworthy or a colluder. 2) the malicious behavior of colluders such as the issue of wrong reports becomes invisible when they successfully form an illegal cluster, i.e., we are assuming that colluders in an illegal cluster behaves as an honest peer.

7. Concluding Remarks

In this paper, we propose two colluder detection schemes for P2P CDNs. The first scheme maintains the reputation score of each peer by continuously collecting reports from peers participating in the system, and the second scheme improves the quality of the colluder detection in the first scheme by using a technique used in the system level diagnosis. The simulation results show that the proposed schemes could certainly detect all colluders without making misidentification provided that the number of colluders is bounded by 10% of the participants.

Possible topics for future work is summarized as follows:

- We need to evaluate the performance of the proposed schemes under more practical situations, for example, a colluder probabilistically conducts malicious action or it follows a specific strategy besides the growing factor.
- The proposed schemes certainly detect suspected peers, but there is no guarantee that those peers are actually colluders. Thus, we need to design a verification scheme to increase the accountability of the overall scheme.

Acknowledgements

This work was supported in part by the Scientific Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan and the Telecommunications Advancement Foundation.

References

[1] E. Abdullah and S. Fujita, "Prevent contents leaking in P2P CDNs with robust and quick detection of colluders," J. Information Processing (JIP), vol.20, no.2, pp.378–385, April 2012.
[2] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," ACM Computing Surveys, vol.36, no.4, pp.335–371, Dec. 2004.
[3] B. Cohen, "Incentives build robustness in BitTorrent," Proc. 1st Workshop on Economics of Peer-to-Peer Systems, 2003.
[4] M.E. Dick, E. Pacitti, and K. Bettina, "Flower-CDN: A hybrid P2P overlay for efficient query processing in CDN," Proc. 12th Int'l Conf. on Extending Database Technology: Advances in Database Technology (EDBT '09), pp.427–438, 2009.
[5] C. Huang, A. Wang, J. Li, and K.W. Ross, "Understanding hybrid CDN-P2P: Why limelight needs its own Red Swoosh," Proc. 18th Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '08), pp.75–80, 2008.
[6] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support Systems, vol.43, no.2, pp.618–644, March 2007.
[7] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," Proc. 12th Int'l Conf. on World Wide Web (WWW '03), pp.640–651, 2003.
[8] X. Liu, T. Huang, L. Huo, and L. Mou, "A DRM architecture for manageable P2P based IPTV system," Proc. IEEE Conf. on Multimedia and Expo, pp.899–902, July 2007.
[9] Q. Liu, R. Safavi-Naini, and N.P. Sheppard, "Digital rights management for content distribution," Proc. Australasian Information Security Workshop Conference on ACSW frontiers 2003, pp.49–58, 2003.
[10] X. Lou and K. Hwang, "Collusive piracy prevention in P2P content delivery networks," IEEE Trans. Comput., vol.58, no.7, pp.970–983, July 2009.
[11] D.K. Mulligan, J. Han, and A.J. Burstein, "How drm-based content delivery systems disrupt expectations of "personal use"," Proc. 3rd ACM Workshop on Digital Rights Management (DRM '03), pp.77–89, 2003.
[12] B.C. Ooi, C.Y. Liau, and K.-L. Tau, "Managing trust in peer-to-peer systems using reputation-based techniques," Proc. WAIM, pp.2–12, 2003.
[13] F.P. Preparata, G. Metze, and R.T. Chien, "On the connection assignment problem of diagnosable systems," IEEE Trans. Electronic Computers, vol.16, no.6, pp.848–854, Dec. 1967.
[14] P. Rodriguez, S.-M. Tan, and C. Gkantsidis, "On the feasibility of commercial, legal P2P content distribution," ACM SIGCOMM Computer Communication Review, vol.36, no.1, pp.75–78, Jan. 2006.
[15] X. Shangqin, L. Zhengding, L. Hefei, and Z. Fuhao, "A trust scheme based DRM model for P2P system," Wuhan University J. Natural Sciences, vol.11, no.5, pp.1373–1377, Sept. 2006.
[16] A. Sherman, A. Stavrou, J. Nieh, A.D. Keromytis, and C. Stein, "Adding trust to P2P distribution of paid content," Proc. 12th Int'l Conf. on Information Security (ISC '09), pp.459–474, 2009.
[17] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," IEEE Trans. Knowl. Data Eng., vol.16, no.7, pp.843–857, 2004.
[18] 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," Proc. 17th ACM Int'l Conf. on Multimedia (MM '09), pp.25–34, 2009.
[19] Y. Zhang, C. Yuan, and Y. Zhong, "Implementing DRM over peer-to-peer networks with broadcast encryption," Advances in Multimedia Information Processing AI PCM 2007, Lect. Notes Comput. Sci., vol.4810, pp.236–245, 2007.
[20] X. Zhang, D. Liu, S. Chen, Z. Zhang, and R. Sandhu, "Toward digital rights protection in BitTorrent-like P2P systems," Proc. 15th ACM/SPIE Multimedia Computing and Networking (MMCN 2008), vol.6818, p.68180F, 2008.
[21] R. Zhou and K. Hwang. "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," IEEE Trans. Parallel Distrib. Syst., vol.18, no.4, pp.460–473, April 2007.

Ervianto Abdullah received the B.E. degree and the M.E. degree in information engineering from Hiroshima University in 2010 and 2012, respectively. He is currently with NTT Data Corporation. His research interests include parallel and distributed computer systems, especially peer-to-peer contents distribution networks and peer-to-peer reputation systems.

Satoshi Fujita received the B.E. degree in electrical engineering, M.E. degree in systems engineering, and Dr.E. degree in information engineering from Hiroshima University in 1985, 1987, and 1990, respectively. He is a Professor at Graduate School of Engineering, Hiroshima University. His research interests include communication algorithms, parallel algorithms, graph algorithms, and parallel computer systems. He is a member of the Information Processing Society of Japan, SIAM Japan, IEEE Computer Society, and SIAM.