Prevention of Polluted Contents in P2P Content Sharing System

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2011

Hiroaki YAMANAKA

Publications

1. Journal

1. H. Yamanaka, S. Okamura, T. Fujiwara, M. Yoshida, Y. Ishihara, T. Akiyama, S.X. Kato, and S. Shimojo, "Refinements of Raters' Similarity Computation for Prevention of Downloading Polluted Contents," *IPSJ Journal*, vol. 51, no. 8, pp. 1428–1442, Aug. 2010. (in Japanese)

2. International Conference

1. H. Yamanaka, S. Okamura, and T. Fujiwara, "A Metric of Search Efficiency and Preventability of Polluted Contents for Unstructured Overlay," in *2010 10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, pp. 181–184, July 2010.

3. Domestic Conference

1. H. Yamanaka, S. Okamura, S.X. Kato, T. Akiyama, and S. Shimojo, "A Method of Searching for Peers Rating Incredibly in Reputation Systems using Trust Chain", SCIS 2008, 4B1-1, CD-ROM, p. 309, Jan. 2008. (in Japanese)

2. H. Yamanaka, S. Okamura, and T. Fujiwara, "Improving the Performance of a Reputation System by Exclusion of Dishonest Ratings based on Similarity between Raters," IPSJ SIG Technical Report, 2008-DPS-136/2008-GN-69/2008-EIP-41, pp. 59–64, Sep. 2008. (in Japanese)

3. H. Yamanaka, S. Okamura, and T. Fujiwara, "A Study on Metrics of Search Efficiency and Prevention of Polluted Contents for Unstructured Overlay," IPSJ SIG Technical Report, vol. 2010-DPS-142/2010-CSEC-48, no. 12, pp. 1–7, Feb. 2010. (in Japanese)

Abstract

Recently, peer-to-peer (P2P) systems have been investigated and developed widely by not only the academic community, but also the industrial community because P2P systems provide a good substructure for large-scale data sharing, content distribution, and application-level multicast. In P2P systems, each peer acts as both a server and a client, and a logical network called an overlay is formed by peers. Cooperation of peers with communications via links on overlay yields fault tolerance and massive scalability properties.

The P2P content sharing system is the best known P2P application. Because of its openness, growth of popularity, and large interference of legitimate content sharing by the dissemination of polluted contents (e.g., faked or virus injected contents), malicious peers that provide polluted contents may exist. Hence, methods for prevention of polluted contents are necessary.

In this dissertation, the purpose of the research is to achieve preventability of polluted contents as well as search efficiency in the P2P content sharing system. First, we estimate the trustworthiness of peers as content providers using a reputation system. A reputation system computes a trust value, which is the estimation of trustworthiness of a peer, from rating values by other peers. To avoid downloading polluted contents, one peer among peers responding query is selected to be a source of downloading content based on the estimation result of them. There are dishonest peers that try to degrade the reliability of trust values by rating other peers with untrue values. Furthermore, the ratio of dishonest raters to all peers or the probability that a peer rates dishonestly fluctuates because of the open and anonymous nature of the P2P system. Hence, a method of trust value computation should be robust to such dishonest ratings. Among existent methods, simple methods such as the arithmetic average perform best when there is a low ratio of dishonest raters, or there is a low probability of rating dishonestly. Sophisticated methods, on the other hand, use the similarity between the rating values for common ratees by two peers perform best when there is a high ratio of dishonest raters or there is a high probability of rating dishonestly.

This dissertation proposes two methods of trust value computation, which perform best regardless of the ratio of dishonest raters or the probability of rating dishonestly. The first method is for a simple peer behavior. This method computes reliable trust values by inferring the honesty or the dishonesty of rating values wider than existent methods using a chain relation of both equality and inequality of rating values for common ratees. The second method is for a probabilistic peer behavior. The method computes reliable trust value by estimating the probabilities that other peers rate honestly more precisely than existent methods using a chain of peers that have common ratees. Furthermore, when all raters of responding peers still cannot be inferred, both methods compute trust values of responding peers by an arithmetic average to achieve high performance at the low ratio or the low probability of rating dishonestly. By simulations of content sharing, the effectiveness of each method is confirmed.

Secondly, construction of an overlay is considered. Searching for content is established by forwarding query messages through links on an overlay with a response by some of peers receiving a query message. A query message is never forwarded further when the peer responds. Furthermore, the receiving peer may respond to try to provide polluted contents even when the peer does not hold the content matching the query. Therefore, the line-up of responding peers and the number of forwarded query messages largely depend on the position of peers around the peer initiating a query on an overlay. In addition, even if the trustworthiness of a peer can be estimated precisely, a peer initiating query cannot obtain its desired content when no trustworthy peer is included in responding peers. This situation implies that the topology of the overlay is significant; hence, a method of constructing an overlay that achieves both search efficiency and preventability of polluted contents is necessary. Unfortunately, peers where both probabilities of providing honest contents (PHC) and holding desired contents (HDC) are high are only a small part of all peers. Their PHCs and HDCs are varied. Even among honest peers, their PHCs are varied because they provide polluted contents mistakenly. HDCs are also varied because held contents reflect their individual interests. A reasonable solution is that peers with high PHCs are privileged to achieve high performance. However, this privilege is not considered in existent methods of topology adaptation, which is a process for each peer's selection of its neighbors. A method of topology adaptation is proposed so that peers with high PHCs are rewarded. To show the effectiveness of our method, simulations of content sharing with topology adaptation are conducted by comparing some of existent methods.

Contents

Chapter 1

Introduction

1.1 Background of the Research

Recently, peer-to-peer (P2P) systems have been investigated and developed widely by not only the academic community but also the industrial community because P2P systems provide a good substructure for large-scale data sharing, content distribution, and application-level multicast applications. A P2P system is a distributed system composed of networked computers called peers which function symmetrically, i.e., each peer acts as both a server and a client. Hence, a peer is also called a servant. This is opposite to a traditional server-client system. A logical network called an overlay, which is independent from physical network (e.g., the Internet), is formed by peers. Services or resources can be served at multiple sites redundantly or distributively, which brings resource sharing with fault tolerance and massive scalability properties by the cooperation of peers with communications via links on overlay. Moreover, it goes beyond services offered by client-server systems. There are numerous applications of P2P systems. Examples of these applications are content sharing systems (e.g., Napster, Gnutella[1], and BitTorrent[2]), distributed file systems (e.g., CFS [1] and OceanStore [2]), streaming media (e.g., PeerCast [3]), communications networks (e.g., Skype[3] and Windows Live Messenger[4]), etc. P2P technology is attracting much attention with its prospects.

Most P2P applications are usually open so that anyone can join as a peer almost anonymously. P2P applications also lack any hierarchical organization or centralized control. Furthermore, most applications are usually

[1] http://rfc-gnutella.sourceforge.net/
[2] http://www.bittorrent.com/
[3] http://www.skype.com/
[4] http://messenger.live.com/

deployed over the Internet. Hence, malicious users who join as peers and behave selfishly or maliciously can join easily although the cooperation of peers in general is significant for P2P applications. Various kinds of selfish or malicious behavior are investigated, e.g., free-riding on content sharing [4], an Eclipse attack which intercepts the honest peer's communications by surrounding the honest peer with malicious peers on an overlay [5], dishonest processing on forwarding messages on an overlay [6], etc. Provisions for malicious peers are necessary.

P2P content sharing systems (e.g., Napster, Gnutella, and BitTorrent) are the most well known applications of P2P systems. In the application, each peer provides its own content to other peers when requested. A major issue on the application is dissemination of polluted contents (e.g., faked or virus injected contents) [7–9]. It is observed that 80% of popular contents are polluted in real P2P content sharing systems [8]. It is warned that P2P networks will be significant methods of delivery of malicious codes as popularization of P2P systems [7]. Furthermore, legitimate content sharing is largely interfered with by dissemination of polluted contents [9]. Hence, there can be malicious peers that provide polluted contents and preventing dissemination of polluted contents in P2P content sharing systems is necessary.

1.2 P2P Content Sharing System

A P2P content sharing system is a P2P application for sharing content held by peers within them. In a P2P content sharing system, each peer provides their own content to other peers when requested, while each peer downloads its desired content from others. Without loss of generality, a P2P content sharing system employs the function of the search for content. As a fundamental behavior, each peer searches for its desired content using the function when trying to download. The function of searching yields some candidates of peers to request content. The peer selects one peer from the candidates, and downloads content from the selected peer.

Usually, anyone can join a P2P content sharing system easily. Hence, dissemination of polluted contents by malicious peers is apprehensive [7–9]. It is assumed that there are roughly two kinds of peers which are honest or dishonest peers. An honest peer provides honest contents while a dishonest peer provides polluted contents. In addition, it is assumed that peers can percept whether their downloaded contents are honest or polluted. Note that peers provide content downloaded from others (i.e., content is replicated over peers) in most P2P content sharing systems. This dissertation assumes that an honest peer filters held polluted contents and provides only honest

contents while content is replicated.

1.3 Purpose of this Research

The purpose of this research is to achieve preventability of polluted contents as well as search efficiency in the P2P content sharing system. To avoid downloading polluted contents, trust value computation of a reputation system that is robust to dishonest ratings is established. To obtain only honest peers as a result of searching efficiently, construction of an overlay is established.

1.3.1 Establish Trust Value Computation

As mentioned in the preceding section, the honest and dishonest peers are assumed. A peer estimates the trustworthiness of other peers and selects a peer to be a source of downloading content based on the estimation result. The trustworthiness of peers is estimated using a reputation system wherein peers rate each other and their trust values are computed from rating values. Advantages of selecting a trustworthy source are to avoid both useless communication costs by downloading polluted contents, and damage by polluted contents that occurs immediately when such contents has been downloaded.

There can be peers that try to degrade the reliability of trust values by rating untruthfully because P2P content sharing system will be a significant method for malicious users to disseminate polluted contents [7]. In addition, the ratio of the number of malicious peers to the number of all peers in the system is unpredictable because of the openness of the P2P content sharing system. Therefore, methods that compute reliable trust values regardless of the ratio of dishonest raters are necessary.

In this dissertation, methods of trust value computation are proposed. To concentrate on estimation of the trustworthiness of peers, any concrete search method and overlay are not assumed here. Instead, it is assumed that candidates to be requested content are given by a certain search method. Two models of peer behavior are assumed. The first model is simple peers that always rate either honestly or dishonestly and always provide either honest or polluted contents. For the simple peer model, rating values by a dishonest rater can be interpreted reversely. Hence, for the model, a high performance method, which infers the honesty of rating values widely and computes trust values using the reverse interpretation, is proposed. The other peer behavior model is probabilistic peers that rate dishonestly and provide polluted contents with probabilities. The probabilistic peers confuse reputation systems

[10, 11]. For the probabilistic peers, precise estimation of the probability that a peer rates honestly is significant for trust value computation; hence, another high performance method, which estimates precise probability of rating honestly by other peer, is proposed. By simulations of content sharing system, the effectiveness of the proposed methods is confirmed.

1.3.2 Establish Overlay Construction

To search for content, an overlay is formed by peers. Regardless of methods of forwarding query messages or structures of overlays, a query message for searching travels though a path on an overlay. If a peer receiving a query message responds, the query message never travels in most search methods. Naturally, a peer responds when it has the contents matching the received query message. In addition, a dishonest peer responds to try to provide polluted contents although it does not hold the contents matching the query [9, 12].

Therefore, the line-up of responding peers and the number of links on an overlay through which a query message travels largely depends on the position of peers on the overlay. Even if the trustworthiness of peers can be estimated precisely by a reputation system, a peer initiating a query cannot obtain its desired content when no trustworthy peer is included in responding peers. This situation implies that the topology of the overlay is significant for both preventability of polluted contents and search efficiency. Therefore, a method of construction of an overlay that achieves both search efficiency and the preventability of polluted contents is necessary.

Overlays are classified into two types that are structured or unstructured. Structured overlays [13–16] have the restriction on the position of peers by peer's ID that is irrelevant to peer behavior. On the other hand, unstructured overlays have no restriction on the position of peers. Hence, unstructured overlays are suitable to design a construction method.

To realize construction of an overlay by a centralized system which tells neighbors on the overlay to each peer, it costs greater than existent centralized systems in P2P content sharing system. The existent centralized systems store information about each peer's held contents or holding peers for contents, and tells about peers that hold requested contents, e.g., the index server of Napster or the tracker of BitTorrent. On the other hand, the centralized system of construction of an overlay additionally has to store information about the trustworthiness of each peer as a provider to construct an overlay with both the preventability of polluted contents and search efficiency; hence, construction of an overlay by a centralized system is confronted with limitation on scalability that is more severe than the existent centralized

systems in P2P content sharing system. A method of overlay construction in distributed manner is necessary.

Methods of overlay construction in distributed manner are proposed in previous works [12, 17–22]. Following these previous works, a method of topology adaptation, which is a process executed by each peer to select its neighbors to construct unstructured overlay with the preventability of polluted contents and efficient search, is proposed. By simulations of content sharing with topology adaptation, the effectiveness of the proposed method is confirmed.

1.4 Outline of the Dissertation

This dissertation is composed of 5 chapters. The remainder of the chapters is organized as follows. First, in Chapter 2, a content sharing system with a reputation system is described. In addition, existent methods of trust value computation are reviewed, and a method of trust value computation is proposed. This method of trust value computation achieves high performance regardless of the ratio of dishonest raters where the simple model of peers that always provide either honest or polluted contents while always rate either honestly or dishonestly. Furthermore, simulations of content sharing varying the ratio of dishonest raters are conducted and the effectiveness of the proposed method is shown.

In Chapter 3, a method, which achieves high performance regardless of the ratio of dishonest raters or the probability of rating dishonestly, is proposed where any peer is assumed to provide polluted contents and rate dishonestly with individual probabilities. In addition, simulations varying the ratio of dishonest raters and the probability of rating dishonestly are conducted and the effectiveness of the proposed method is shown.

In Chapter 4, content sharing on an unstructured overlay and a framework of topology adaptation are described. Furthermore, existent methods of topology adaptation are reviewed and a method is proposed so that peers with a high probability of providing honest contents achieve both an efficient search and preventability of polluted contents. The proposed method is evaluated by simulations of topology adaptation and its effectiveness is shown.

Finally, in Chapter 5, the contributions of this dissertation are summarized and future works are described.

Chapter 2

Trust Value Computation for Simple Peers

2.1 Introduction

When a peer obtains its desired content, the peer first initiates a query. Next, some of peers receiving query respond to the initiating peer and the initiating peer selects a peer from the responding peers and downloads content from the selected peer. The reputation system [10, 11, 19, 23–27] can be applied to avoid downloading polluted contents. The reputation system computes the trust values of peers, which is an estimation of their trustworthiness. The reputation system is composed of peers and a database of rating values. In this system, peers play the two roles. One role is as a rater that registers the rating value to the database when the rater has downloaded content. The other role is a ratee that is rated (i.e., the rating value for the ratee is registered), when the ratee has provided content for the rater. The rating value is computed by the rater based on the perception of whether the downloaded contents are honest or polluted, and the rating value is registered in the database. Trust values of peers are computed from the registered rating values and used to select a peer from responding peers to avoid downloading polluted contents in the future.

Because a P2P content sharing systems is a significant method to disseminate polluted contents, malicious peers may be dishonest raters that try to degrade the reliability of trust values by registering untrue rating values. Furthermore, the ratio of the number of dishonest raters to all peers may fluctuate largely. The ratio can grow more than 40% [10] by the Sybil attack [28] where a single malicious user obtains multiple identifiers. To prevent the Sybil attack, an identity authority should be employed. However, such

an authority is usually absence in P2P content sharing system due to limitation on scalability, large costs or responsibility for administration of the authority. Hence, the ratio of dishonest raters of more than 40% should be considered. However, growing the ratio of dishonest raters ultimately is impractical. This dissertation assumes that the ratio of dishonest raters is 50% at most. Therefore, a method that computes reliable trust values regardless of the ratio of dishonest raters that is not more than 50% is necessary. In this chapter, the simple peer behavior is assumed. The simple peer, as a rater, always rates either honestly or dishonestly, and provides either honest or polluted contents as a provider.

Among existent methods [10, 11, 19, 23–27, 29–33], when the ratio of dishonest raters is low, a simple method such as trust value computation by an arithmetic average [19, 32] performs best. On the contrary, when the ratio of dishonest raters is high, the method using a similarity between the rating values for common ratees by themselves and other peers, called the raters' similarity, as the weight for rating value by the peers [11, 23, 26] performs best while the computation method by arithmetic average performs considerably less well. The method using the raters' similarity is not deteriorated by the ratio of dishonest raters while the method by arithmetic average is deteriorated directly. However, because the methods using the raters' similarity set raters' similarity to zero if the peer and the rater do not have a common ratee, trust values cannot be defined, and the providing peer has to be selected randomly when there are frequently no common ratees with any other peers. Hence, the method by the arithmetic average performs better. This often occurs when the peer joins in the system.

In this chapter, a method, which achieves high performance regardless of the ratio of dishonest raters for the simple peer behavior, is proposed. Since the simple peer behavior is assumed, rating values by other raters can be inferred to be either honest or dishonest. The rating values inferred to be honest are straightforwardly the trust values of the ratees while the rating values inferred to be dishonest are the reverse of the trust values of the ratees. Our method infers the honesty of rating values by other peers by examining the rating values for a common ratee. It is the same as the existent methods which use the raters' similarity as the weights [11, 23, 26]. Moreover, our method infers the honesty of rating values more widely than the existent methods [11, 23, 26] using a chain relation of both equality and difference of rating values. The rating values are for common ratees between peers along a chain of peers that have common ratees. Furthermore, if the honesty of all the raters of the responding peers cannot be inferred, then the proposed method computes trust values by arithmetic average to achieve a high performance as good as the existent methods of arithmetic average [19, 32] especially when

Table 2.1: Significant parameters.

P	The set of all peers
C	The set of shared honest contents
P_c	The set of candidates from which content c is downloaded
rt_{jx}	Rating value for peer x by peer j
tv_{ix}	Trust value of peer x computed by peer i

the ratio of dishonest raters is low. To show the effectiveness of the proposed method, simulations of content sharing are conducted by comparing known methods.

The remainder of this chapter is as follows. First, Section 2.2 describes a content sharing system with a reputation system and its performance metric. Section 2.3 describes a framework of trust value computation and reviews existent methods. Sections 2.4 and 2.5 describe assumed simple peer models and our proposal for the model. Section 2.6 evaluates our proposal by simulation. Finally, Section 2.7 concludes this chapter.

2.2 Content Sharing System with Reputation System

In this section, a content sharing system with a reputation system and the performance metric for the system is described. Note that the description in this section is also referred to in Chapter 3.

2.2.1 Description of the System

In this dissertation, a P2P content sharing system that applies a reputation system to avoid polluted contents is considered. Here, this system is described. Table 2.1 shows the significant parameters in our description. Note that systems in previous works [10, 11, 19, 23–27] also can be described as follows.

Let P be the set of all peers and C be the set of all honest contents shared by the peers. In this dissertation, P and C are constant for simplicity. Any content in C is owned by one or more peers in P and is provided by the peers. A fraction of peers in P hold polluted contents and provide such contents. It is assumed that any peer in P can perceive immediately whether the

downloaded contents are honest or polluted, i.e., the downloaded content is in C or not.

Generally, a reputation system is composed of all peers in P and a database of rating values, and has functions for publishing rating values and computing trust values as follows:

- Rating values rt_{jx} for any pairs of peers $j, x \in P$ are stored in the database. The values of rt_{jx} ($x \in P$) are computed and registered by peer j. Peer j has information inf_j, and the function g for computing rating values from the information is defined. Peer j obtains the value of rt_{jx} by $rt_{jx} = g(x, inf_j)$. Note that, if peer j is a dishonest rater, peer j not always registers rating values by the function g. Usually, in a P2P content sharing system, the database of rating values is a distributed system where each peer stores and publishes a fraction of all rating values. It is assumed that published rating values are reliable by digital signatures of the registering peers. Note that digital signatures using peers' self-signed certificates and using hash values of public keys as peers' IDs [34] can achieve a practical anonymity because a certificate authority (CA) is absence, and the IDs are not associated with users.

- Let M_{rt} be the matrix $M_{rt} = (rt_{jx})$. The function f for computing trust values of any peers from M_{rt} is defined. Let tv_{ix} be the trust value of peer x for peer i. The value of tv_{ix} is obtained by $tv_{ix} = f(i, x, M_{rt})$. Note that a trust value is a real number and peer i interprets $tv_{ix} < tv_{ix'}$ as peer x' as more trustful as a content provider than peer x.

An example of definition of rt_{jx} is the ratio of the number of peer j's downloads of honest contents from peer x to the number of all peer j's downloads from peer x. In this example, inf_j has to contain the numbers of downloads. It is assumed that if peer j has never downloaded from peer x, rt_{jx} is stored and published as the value rt_{init}.

By applying the reputation system described immediately above, in content sharing system, peer $i \in P$ operates as follows when it tries to download content $c \in C$.

(i) Peer i obtains some responding peers as a result of a certain method of searching for content. Let P_c ($\subseteq P$) be the set of responding peers except for the peer that had provided polluted contents when peer i queried about content c before.

(ii) Peer i computes trust value tv_{ix} of each peer x in P_c and selects one peer $x_0 \in P_c$ whose trust value is the highest among peers in P_c. Note that if there are multiple peers whose trust values are equal and the highest,

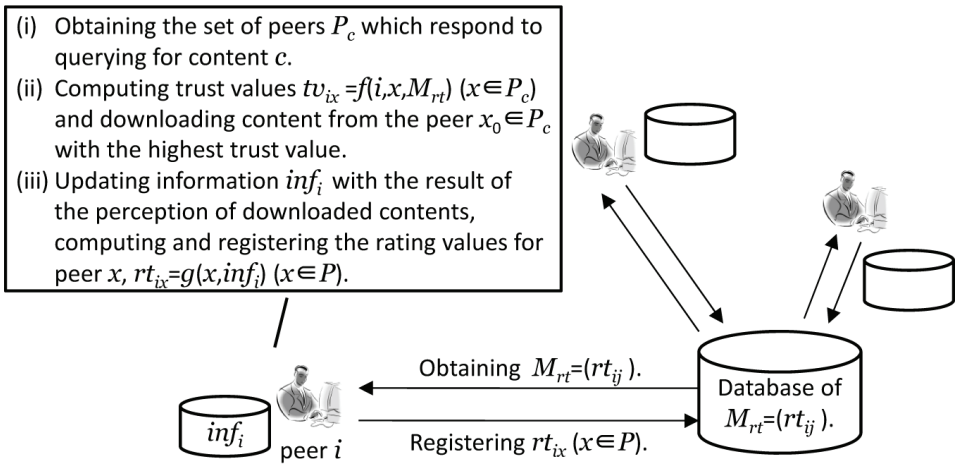


Figure 2.1: The composition of the reputation system.

peer i selects one peer randomly from the multiple peers. Then, peer i requests the selected peer x_0 to provide content c and downloads it.

(iii) Peer i updates inf_i based on the result of perception, i.e., whether the downloaded contents is honest or polluted. Then, peer i computes rating values rt_{ix} $(x \in P)$ by the function $g(x, inf_i)$ and registers them to the database.

The composition of the reputation system with peer's processing are shown in Figure 2.1. This dissertation assumes that a malicious peer responds to provide polluted contents although it does not hold the queried contents. Hence, a responding peer at (i) is a peer that holds content c or a peer that does not hold content c but provides polluted contents.

As described in Section 1.3, any concrete search method and overlay are not assumed in Chapters 2 and 3. Instead, it is assumed that responding peers are obtained by a certain method in the ratio of the numbers of peers that provide honest contents and peers that provide polluted contents over all peers.

2.2.2 Performance Metric of the System

From the description of the system in Section 2.2.1, the designs of functions f and g, and inf_i are significant for the content sharing system. The design of function f is particularly significant. The performance metric of the content sharing system must be determined to design the system. The performance

metric should indicate the effect of the prevention of polluted contents where each peer tries to obtain its desired content, since prevention of polluted contents is our objective.

There are some metrics in previous works [10, 11, 19]. For example, in PeerTrust PSM [11], the performance metric is the difference between the ideal and computed trust values of each peer. This metric is not suitable for our purpose since it does not indicate the effect of prevention of polluted contents. In [10], the metric is the ratio of the number of downloads of polluted contents to the number of all downloads. In [19], the metric is the number of downloads of polluted contents after a certain number of downloads. Both of the metrics [10, 19] indicate the quantity of downloads of polluted contents. However, they do not reflect the situation where each peer tries to obtain its desired content. The line-up of contents that each peer desires naturally varies. Hence, the numbers of desired contents are varied. Nevertheless, the metrics [10, 19] assume that all peers download equal times.

Contrary to previous works, it is assumed that each honest peer repeats downloads until it obtains all of its desired content. Moreover, our metric is the ratio of the number of downloads of polluted contents to the number of all downloads of honest contents, which is equal to the number of the peer's desired contents. Intuitively, this ratio is the mean number of eventual downloads of polluted contents between two downloads of honest contents. Hence, the ratio is called the rate of downloading polluted contents. Note that the performance of the system is measured by the average of the rates of downloading polluted contents over all honest peers. It is interpreted that the lower the averaged rates, the higher the performance of the content sharing system used with the reputation system.

2.3 Design of Trust Value Computation

This section describes the framework of trust value computation. Performance against the ratio of the number of peers that rate dishonestly is a matter of interest when designing trust value computation. Inferring the reliability of rating values in computing trust values is significant to compute reliable trust values with handling this matter. Hence, related works are described with respect to the weight of rating values. Note that this section is also referred to in Chapter 3.

2.3.1 Computing Trust Values

In previous methods of computing trust values [10, 11, 19, 23–27, 29–33], reliability of rating values is estimated and rating values are weighted based on the estimation. Let wt_{ij} be the weight for the rating values rt_{jx} ($x \in P$) by peer $j \in P$ that peer i uses when computing trust values. The $|P|$ dimensions vector $\mathbf{w}_i$ is defined as

$$\mathbf{w}_i = (wt_{i1}, wt_{i2}, \cdots, wt_{i|P|}).$$

The subscripts of the weight are the trust value calculator i and the rater j, since, in our design, the weight for rt_{jx} ($j, x \in P$) used when computing tv_{ix} depends only on the peers i and j, which is the same in [10, 11, 19, 23–27, 29, 31, 32]. Note that the weight can depend on the object of computing trust value, the peer x, in addition to the peers i and j, which is the same in [30, 33].

The function $f'(i, M_{rt}) = \mathbf{w}_i$, which calculates the vector $\mathbf{w}_i$ from matrix M_{rt}, is defined in [10, 11, 19, 23–27, 29–32]. Furthermore, the function f'' is defined to compute trust values from weights $\mathbf{w}_i$ and rating values M_{rt},

$$f''(\mathbf{w}_i, M_{rt}) = (tv_{i1}, \cdots, tv_{i|P|}). \tag{2.1}$$

In practice, computing trust values of only the peers in P_c is sufficient to select a source of downloading content. For instance, f'' is defined by the weighted average of rating values [11, 26, 30], or the maximum value among the products of the weights and the rating values [24], etc. In [33], both of the numbers that peer x has been judged to have provided honest and polluted contents are used when computing the trust value of peer x. In this case, the method can be corresponded to our description by slight modification; inf_j contains the two values and inf_j is one of the inputs of f''.

2.3.2 Related Works

In this section, we review previous methods, especially, the design of f'', i.e., the calculation of the weight of rating values. Note that the system applying the reputation system in [32] is a general P2P resource sharing system and that in [29–31, 33] is a multi-agent system. However, the methods of computing trust values in these methods can be applied readily to the P2P content sharing system.

Methods without countermeasure against dishonest raters In STEP [19] and [32], the trust value of a peer is computed by the sum or arithmetic average of the rating values for the peer. In these methods, all

rating values are dealt with equally; hence, all elements of $|P|$ dimensions vectors $\mathbf{w}_i$ ($i \in P$) are 1. Therefore, there is no countermeasure against dishonest raters and the reliability of computed trust values is degraded when the ratio of dishonest raters to all the raters is high.

Methods using statistical analysis of rating values In [33], the reliability of a rating value for peer x is estimated by the mean of Beta distribution where the numbers of downloaded content from peer x is judged to be honest and to be polluted are its parameters. Each of the numbers is the sum of judgments by all peers that download content from peer x. If the rating value for peer x by peer j largely differs from the mean, all the rating values by peer j are regarded as dishonest and excluded when computing trust values. Hence, the weight for the rating values by peer j is 1 if regarded as honest, or 0 if regarded as dishonest. In [30], the weight for rating value by one peer is determined by the Chi-square test of the rating values by the peer and the other peers. In these methods [30, 33], the weight for rt_{jx} largely depends on the distribution of rating values for peer x by all peers including peer j. Therefore, similar to [19, 32], the reliability of computed trust values is degraded when the ratio of dishonest raters to all the raters is high [35].

Methods assuming consistency in peer's trustworthiness In EigenTrust [10], it is assumed that a peer's trustworthiness as a provider consists within its trustworthiness as a rater. At initial trust value calculation, the weights for all the rating values are equal. At succeeding calculations to update trust values, the higher the trust value of a peer, the higher the weight of rating values by the peer. Some known methods [24, 27, 29, 31] use the same assumption in EigenTrust [10]. Generally, a peer's trustworthiness as a provider is not always synonymous with its trustworthiness as a rater; hence, these methods determine the weights incorrectly for rating values. For example, if there are peers that provide polluted contents and rate dishonestly, rating values by the peers are weighted largely although the peers are dishonest raters. As a result, the reliability of computed trust values is deteriorated [35].

Methods using rating among raters In [25], a peer rates other peers with respect to rating for other content providers. The weight for rating values by a peer is determined by the number of all positive ratings by other raters. In this method, dishonest rating among raters is not considered since the weight is computed by simple summation. Therefore, if dishonest raters also dishonestly rate other raters, similar

to [19, 32], the reliability of computed trust values is degraded when the ratio of dishonest raters to all the raters is high.

Methods using raters' similarity In PeerTrust PSM [11], P2PRep [23], and [26] a peer computes the weights for rating values by another peer by similarity between the rating values by the peer and the other peer for common ratees. wt_{ij} is the value that represents similarity between the rating values by peers i and j for the common ratees of the two peers.

In the methods using raters' similarity [11, 23, 26], the estimation of the reliability of the rating value by another peer (i.e., the weight) is independent from the trustworthiness of the rater peer as a content provider since the estimation is by similarity of the rating values for common ratees by itself and the other peer. Therefore, the performance of the methods is never deteriorated by an inconsistency of the peer's trustworthiness as a rater and a provider. This is contrary to the methods [10, 24, 27, 29, 31] which assume consistency of trustworthiness of a peer as both a rater and a provider. Furthermore, the estimation of rating values is never affected by the ratio of dishonest raters to all the raters. This is contrary to [30, 33].

However, when there are no common ratee, the methods [11, 23, 26] cannot estimate the trustworthiness of peers as raters because all the weights are defined as zero and subsequently all the trust values are undefined in this circumstance. Hence, a peer has to chose a peer randomly from responding peers. This occurs frequently when peers join the system. On the other hand, the methods [19, 32] can compute trust values regardless of the existence of common ratees because of the trust values computed by the arithmetic average of rating values, etc. Furthermore, in the methods, the reliability of computed trust values is sufficient when the ratio of dishonest raters is 1/3 and less, while the reliability is insufficient when the ratio is more than 1/3 [29].

Therefore, among existent methods, computing by the arithmetic average is most reliable when the ratio of dishonest raters is low. Computing using raters' similarity, in contrast, is most reliable when the ratio of dishonest raters is high.

2.4 The Simple Peer Model

In a content sharing system applied reputation system, a peer plays the role of rater when downloading content and ratee when providing content. In this

chapter, it is assumed that peers behave simply. Any peer always rates either honestly or dishonestly, and provides either honest or polluted contents.

Section 2.2.1 describes how peer $i \in P$ updates information inf_i by the result of perception (i.e., whether downloaded contents are honest or polluted) to compute rt_{ij} ($j \in P$) at each download. In this simple peer model, a rater is either honest or dishonest. If peer i is a dishonest rater, peer i always updates inf_i incorrectly, i.e., peer i registers the opposite to the result of perception. If peer i is an honest rater, it always updates inf_i correctly.

The other aspect of a peer is as a content provider. Section 2.2.1 also describes how peer x provides content that is either content $c \in C$ or polluted contents when peer $x \in P$ is requested to provide content c. In the simple peer model, a content provider is either honest or dishonest. If peer x is an honest provider, it always provides honest contents. If peer x is a dishonest provider, it always provides polluted contents.

2.5 Proposed Method for the Simple Peer Model

2.5.1 Design of Proposed Method

In this section, a trust value computation that infers the honesty of rating values for the simple peer model is proposed. Since the simple peer model is assumed, only three values are enough for rating values and weights. The three values are to represent "honest", "dishonest", and "unknown", respectively (see Section 2.5.2).

As described in Section 2.3.2, the weight of the rating value by the raters' similarity [11, 23, 26] is not affected by the ratio of dishonest raters. However, when the ratio of dishonest raters is low, the method using the raters' similarity performs at less than the arithmetic average [19, 32]. The cause of this is that the value of the raters' similarity for two peers is set to zero when there is no common ratee between the two peers as detailed in Section 2.3.2. To achieve high performance regardless of the ratio of dishonest raters, the proposed method uses raters' similarity as weights, and moreover, the proposed method computes weights wider than the existent methods [11, 23, 26] by inferring the honesty between two peers that have no common ratee. Furthermore, the weight of rating values by other peers that have no common ratee is computed using a chain relation of both equality and difference of rating values. The rating values are for common ratees between peers along a chain of peers that have common ratees (see Section 2.5.3).

Trust values are computed from weights and rating values by the function f'' as in Equation (2.1). Since the simple peer model is assumed, a rating

value for which the weight is "honest" is straightforwardly the trust values for the ratee of the rating value. On the other hand, the rating value for which the weight is "dishonest" is the reverse of the trust value for the ratee of the rating value. Furthermore, when all the weight of rating values for responding peers are "unknown", trust values are computed by an arithmetic average. At this time, one peer is selected based on trust values by the arithmetic average. Our method achieves high performance and is not inferior to computing by arithmetic average [19, 32] when the ratio of dishonest raters is low. Note that since our method infers the reliability of rating values widely, peer selection by trust values computed by the arithmetic average do not occur frequently. Hence, the deterioration will be little (see Section 2.5.4).

2.5.2 Specifications of Rating Values and Weights

Three values are necessary for rating values and weights. Here, $rt_{\text{init}}, 1, -1$ are used to represent "unknown", "positive", and "negative" respectively. Let peer $j \in P$ be an honest rater. If peer j has perceived the downloaded contents from peer $x \in P$ to be honest contents, $rt_{jx} = 1$. If peer j has perceived the downloaded contents from peer x to be polluted contents, $rt_{jx} = -1$. If peer j is a dishonest rater, rt_{jx} takes the opposite value. When peer j has not yet rated peer x (i.e. peer j has not yet downloaded content from peer x), $rt_{jx} = rt_{\text{init}}$ regardless of peer j being an honest or dishonest rater.

Since the simple peer model is assumed, only three values which represent "unknown", "positive", and "negative" respectively are necessary. When computing trust values, peer i infers the honesty of rating values by peer j. If peer i infers peer j to be an honest rater, $wt_{ij} = 1$. On the other hand, if peer j is inferred to be a dishonest rater, $wt_{ij} = -1$. If peer i cannot infer the honesty of peer j as a rater, $wt_{ij} = 0$.

2.5.3 Inferring Honesty of Another Rater

Before computing trust values, the honesty of rating values by another peer is inferred. As the philosophy of the existent methods that use the raters' similarity as the weight [11, 23, 26], each peer assumes that itself is an honest rater. Let peer i infer the honesty of rating values by peer $j \in P$, i.e., peer i computes wt_{ij}. For peer x, let $R_x = \{j \in P : rt_{jx} \neq rt_{\text{init}}\}$ be the set of peers that are raters of peer x. Let $CR_{ij} = \{x \in P : i, j \in R_x\}$ be the set of common ratees between peers i and j. The simple peer model yields the following proposition.

Proposition 1. *Rating values by any two peers $i, j \in P$ of the simple peer*

model are always equal or different for any peer of the simple peer model in P rated by both peers i and j.

Proof. Let peer x be rated by peers i and j, i.e., both the values of rt_{ix} and rt_{jx} are not rt_{init}. Peer x is either an honest or a dishonest provider since the simple peer model is assumed. In addition, peers i and j are honest or dishonest raters. Hence, the honesty as raters of peers i and j either agrees or disagrees.

By the definition of the simple peer model and the specifications of rating values described in Section 2.5.2, if both peers i and j are either honest or dishonest raters, both values of rt_{ix} and rt_{jx} for peer x are either 1 or -1, and vice versa. In a similar way, if peer i is an honest rater while peer j is a dishonest rater or it is reversed, the values of rt_{ix} and rt_{jx} for peer x are different, and vice versa. □

Peer i infers the honesty of peer j as a rater using Proposition 1. By the philosophy that each peer assumes that itself is an honest rater, peer i infers peer j to be an honest rater (i.e., $wt_{ij} = 1$) when peer i can observe that the rating value for a common ratee between peers i and j are equal. On the other hand, peer j is inferred to be a dishonest rater (i.e., $wt_{ij} = -1$) when peer i can observe that the rating value for a common ratee is different.

Inferring when $CR_{ij} \neq \emptyset$

Suppose $CR_{ij} \neq \emptyset$. By Proposition 1, if $rt_{ix} = rt_{jx}$ for peer $x \in CR_{ij}$, it is always true for any other peer in CR_{ij}, moreover, it is true for any peer whom rated by peers i and j in the future. On the other hand, if $rt_{ix} \neq rt_{jx}$ for peer $x \in CR_{ij}$, the same holds. Therefore, if $rt_{ix} = rt_{jx}$, peer i infers that peer j is an honest provider since each peer assumes that itself is an honest rater. On the other hand, if $rt_{ix} \neq rt_{jx}$, the peer i infers that the peer j is a dishonest provider. wt_{ij} $(j \in P, CR_{ij} \neq \emptyset)$ is computed as

$$wt_{ij} = \begin{cases} 1 & (rt_{ix} = rt_{jx} \text{ for a peer } x \in CR_{ij}) \\ -1 & (rt_{ix} \neq rt_{jx} \text{ for a peer } x \in CR_{ij}) \end{cases} . \tag{2.2}$$

Inferring when $CR_{ij} = \emptyset$

Suppose $CR_{ij} = \emptyset$. Even when $CR_{ij} = \emptyset$, wt_{ij} may be computed from wt_{ik} and wt_{kj} using the following proposition.

Proposition 2. *For any three peers $i, j, k \in P$ of the simple peer model, where wt_{ik} and wt_{kj} whose values are 1 or -1 are given and these values are assumed to be correct, $wt_{ij} = wt_{ik} * wt_{kj}$.*

Proof. Since the value of wt_{ik} is assumed to be correct, $wt_{ik} = 1$ (-1) means that rating values for common ratees between peers i and k are equal (different). For $wt_{kj} = 1$ (-1), the same holds. In addition, by Proposition 1, rating values for common ratees between peers i and j in the future are always equal or different. There are four cases of combinations of values of wt_{ik} and wt_{kj}.

- $wt_{ik} = 1, wt_{kj} = 1$
  Rating values for common ratees between peers i and k are equal. Rating values for common ratees between peers k and j are equal. Therefore, rating values for common ratees between peers i and j are also equal; $wt_{ij} = 1 = wt_{ik} * wt_{kj} = 1 * 1$.

- $wt_{ik} = -1, wt_{kj} = -1$
  Rating values for common ratees between peers i and k are different. Rating values for common ratees between peers k and j are different. Therefore, rating values for common ratees between peers i and j are also equal; $wt_{ij} = 1 = wt_{ik} * wt_{kj} = (-1) * (-1)$.

- $wt_{ik} = 1, wt_{kj} = -1$
  Rating values for common ratees between peers i and k are equal. Rating values for common ratees between peers k and j are different. Therefore, rating values for common ratees between peers i and j are also equal; $wt_{ij} = -1 = wt_{ik} * wt_{kj} = 1 * (-1)$.

- $wt_{ik} = -1, wt_{kj} = 1$
  In this case, $wt_{ij} = -1 = wt_{ik} * wt_{kj} = (-1) * 1$. This is proofed alike in the case of $wt_{ik} = 1, wt_{kj} = -1$.

Therefore, $wt_{ij} = wt_{ik} * wt_{kj}$. □

Peer i can infer the honesty of peer j $(CR_{ij} = \emptyset)$ as a rater using Proposition 2 if only there exists at least one sequence $\mathrm{p}_{ij} = (p_0, \cdots, p_d)$ of finite length, where $d \geq 2$, $p_e \in P$ $(0 \leq e \leq d)$, $p_0 = i$, $p_d = j$, $p_e \neq p_f$ if $e \neq f$, and $CR_{p_e p_{e+1}} \neq \emptyset$ $(0 \leq e \leq d-1)$. Weights $wt_{p_e p_{e+1}}$ $(e = 0, \cdots, d-1)$ are obtained by Equation(2.2) since $CR_{p_e p_{e+1}} \neq \emptyset$. Then, wt_{ij} is obtained by applying Proposition 2 recursively. Note that the values of wt_{ij} obtained by any sequences such as p_{ij} are invariable because of the simple peer model. When there is no such sequence between peers i and j, $wt_{ij} = 0$ which means that peer i cannot infer the honesty of peer j as a rater. Hence, wt_{ij} for $CR_{ij} = \emptyset$

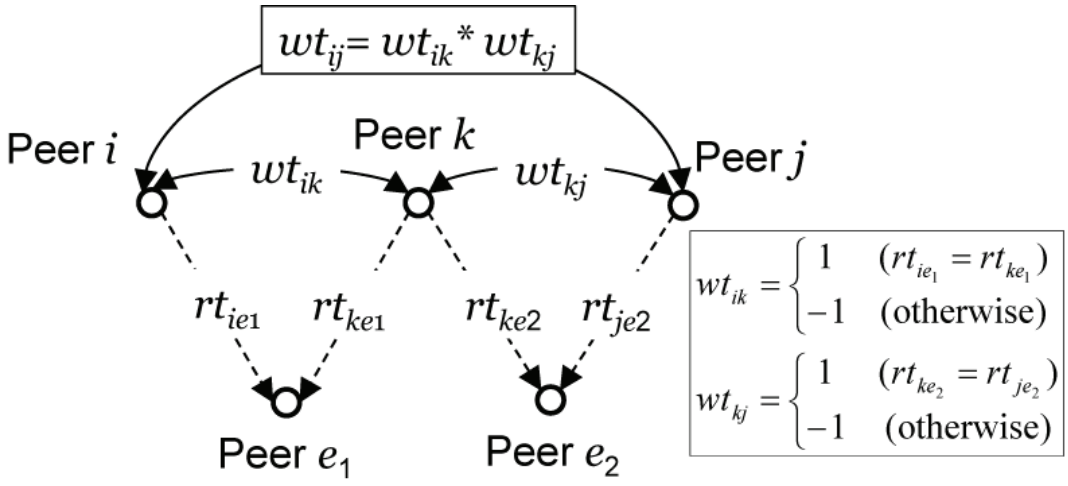


Figure 2.2: An example of computing wt_{ij}.

is computed as

$$wt_{ij} = \begin{cases} \prod_{e=0}^{d-1} wt_{p_e p_{e+1}} & (CR_{ij} = \emptyset \text{ and there exists sequence } \mathrm{p}_{ij}) \\ 0 & (CR_{ij} = \emptyset \text{ and there does not exist sequence } \mathrm{p}_{ij}) \end{cases}.$$

An example of computing weights wt_{ij} using wt_{ik} and wt_{kj} given by existing common ratees is depicted in Figure 2.2.

2.5.4 Computing Trust Values

Peer i has established inferring the honesty of other raters $j \in P \setminus \{i\}$. Next, peer i computes trust values tv_{ix} ($x \in P_c$). Since the simple peer model is assumed, the rating value $rt_{j_0 x}$ by peer j_0 whom peer i infers to be an honest rater (i.e., $wt_{ij_0} = 1$) is straightforwardly the trust value tv_{ix}. On the other hand, the rating value $rt_{j_1 x}$ by peer j_1 whom peer i infers to be a dishonest rater (i.e., $wt_{ij_1} = -1$) is the reverse of the trust value tv_{ix}. If all raters in R_x cannot be inferred, tv_{ix} is set to "undefined" here. Hence, tv_{ix} is computed as

$$tv_{ix} = \begin{cases} wt_{ij} * rt_{jx} & (\text{There exists a peer } j \in R_x \text{ s.t. } wt_{ij} \neq 0) \\ \text{undefined} & (\text{otherwise}) \end{cases}. \quad (2.3)$$

Note that for any peer j in R_x, the value of $wt_{ij} * rt_{jx}$ is either 1 or -1, moreover, the value is invariable because of the simple peer model.

If there are one or more peers in P_c whose trust values are 1, one peer among them is selected to request content c. However, all the trust values can be -1 or "undefined". As mentioned in Section 2.2.1, it is assumed that the responding peers are in the ratio of honest and dishonest providers over all peers. Moreover, it is meaningless that the situation where all peers in P are dishonest providers; hence, all of the trust values of peers in P_c are not necessarily -1. The situation, where there is at least one trust value of "undefined" of peers in P_c, but there is no trust value of 1 of peers in P_c, is considered.

Since a peer whose trust value is -1 is inferred to be a dishonest provider by peer i, peer i tries to select a peer from the peers whose trust values are "undefined". Peer i computes the trust values of the peers whose trust values are "undefined" by Equation (2.3) using an arithmetic average. For peer $y \in P_c$ where tv_{iy} is undefined by Equation (2.3), tv_{iy} is recomputed as

$$tv_{iy} = \begin{cases} \dfrac{|\{j \in R_y : rt_{jy} = 1\}|}{|R_y|} & (R_y \neq \emptyset) \\ \text{undefined} & (\text{otherwise}) \end{cases} . \tag{2.4}$$

Peer i selects one peer whose trust value is neither "undefined" nor -1 but the highest in P_c. If there are still only peers whose trust values are -1 or "undefined", peer i selects one peer randomly from the peers whose trust values are "undefined".

2.6 Evaluation of Proposed Method

2.6.1 Simulation Settings

The proposed method is evaluated and compared with some existent methods. The evaluation is conducted by the simulation of the content sharing described in Section 2.2.1. The performance metric is the average of the rates of downloading polluted contents for all honest peers as described in Section 2.2.2. Figure 2.3 shows the process of the simulation. In the process, desired contents for each peer is set and each peer shares content until all of the desired contents of all peers are obtained. Hence, content sharing is perpetual during the simulation. Finally, the average of the rates of downloading polluted contents for honest peers is computed.

Compared Methods

As mentioned in Section 2.3.2, the performances of the methods with no countermeasure against dishonest raters [19, 32], the methods using statistical

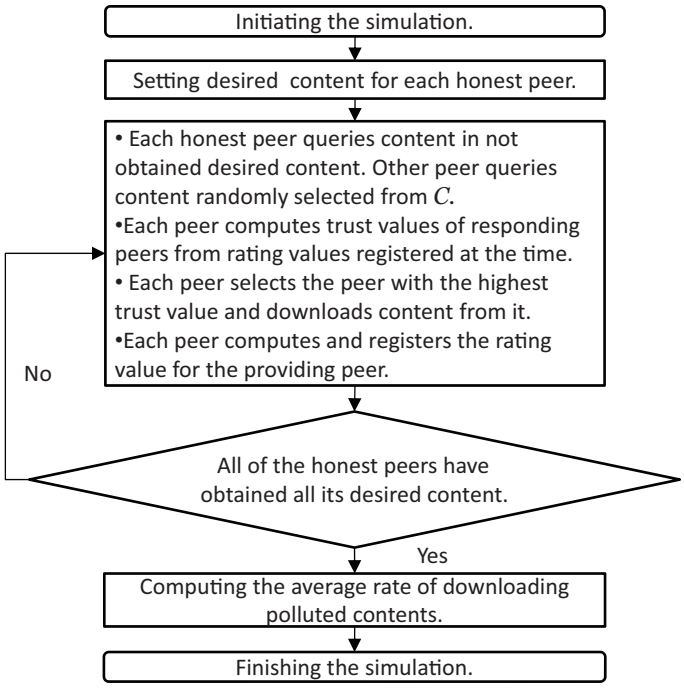


Figure 2.3: The process of the simulation.

analysis of rating values [30, 33], and the method using rating among raters [25] are virtually equal. Hence, among these methods, only the method computing trust values by arithmetic average [32] is compared with the proposed method. From the methods using raters' similarity [11, 23, 26], PeerTrust PSM [11] is compared. EigenTrust etc. [10, 24, 27, 29, 31] are not chosen since it is assumed that the peer's trustworthiness as a provider does not always agree with its trustworthiness as a rater.

Furthermore, to analyze the effectiveness of elements of the proposed method, variances of the proposed method are compared. In the rest of this chapter, the proposed method is called Method A. To evaluate the effectiveness of inference of the reliability of rating values using a chain relation among peers that have a common ratee, variance of the proposed method which infers rating values only by other peers that have a common ratee, i.e., only direct inferring. This variance is called Method A$'$. In addition, to evaluate the effectiveness of computing trust values by arithmetic average, another variance of the proposed method which never computes trust values by arithmetic average, i.e., the trust values being "undefined" by Equation (2.3) are never recomputed by Equation (2.4). This variance is called Method A$''$ in the rest of this chapter.

Table 2.2: Models of peers in the simulation.

Model	Providing	Rating
Honest peer	Always honest contents	Always rating honestly
Malicious peer	Always polluted contents	Always rating dishonestly
Confusing peer	Always honest contents	Always rating dishonestly

Assumed Peer Models

In the simulation, the honest peer, the malicious peer, and the confusing peer are assumed as shown in Table 2.2. The confusing peer confuses the methods of EigenTrust etc. [10, 24, 27, 29, 31] by an inconsistency in the trustworthiness of a peer as a provider and a rater [35]. There can be the converse of the confusing peer, i.e., a peer which always provides polluted contents and rates honestly. However, EigenTrust etc. are not confused by such a peer since peers that provide polluted contents are regarded as dishonest raters and rating values by the peers are discarded when computing trust values. Furthermore, in the existent methods even in the proposed method, the trustworthiness of a peer as a rater is not regarded as having the same trustworthiness as a provider. Therefore, in our simulation, only the three types of peers are picked up.

Settings on Desired and Held Contents

Each honest peer's desired contents are assigned from C by popularity of each piece of content in C. During the simulation, the set C is invariable. In addition, peers that respond a query for each piece of content in C are fixed.

The number of the honest peers that desire each piece of content in C is assumed to be proportional to Zipf distribution on all the ranks of content. It is reported that the number of peers that have each piece of content is distributed in Zipf distribution in practical P2P content sharing system [8]. Note that except for the honest peers, queried content is selected randomly from C for each requesting so that the malicious or confusing peers rates other peers same times.

The ratio of honest and dishonest providers in responding peers is assumed to be the ratio of honest and dishonest providers over all peers as described in Section 2.2.1. Hence, in the simulation, for content in C, 20 different peers are selected randomly form all peers to be the responding peers.

Table 2.3: The default values of parameters in the simulation.

# of all peers $\|P\|$	500
# of all honest contents $\|C\|$	(# of the honest peers)*10
# of experiments over which results are averaged	5

Other Settings

Table 2.3 shows the default values of parameters in the simulations. The number of all honest contents $|C|$ is set to the multiple of the number of the honest peers for the fairness of the number of desired contents per honest peer.

2.6.2 Results

Since assuming the simple peer model, the performance against the ratio of dishonest raters or providers is a significant matter of interest. Hence, in this section, to influence the ratio of the dishonest raters and providers, simulations are conducted by varying the number of the malicious peers. Figure 2.4 shows the average rates of downloading polluted contents against the number of malicious peers. The number of the malicious peers varies between 25 (5%) and 250 (50%) of 500 while that of the confusing peers is set to 50 (10%) of 500. The honest peers form the reminder.

As shown in Figure 2.4, Method A (the proposed method) performs best except for the case when the number of the malicious peers is 250 (50%). Moreover, the results show that elements of the proposed method (i.e., inference of the reliability of rating values using a chain relation and computing trust value by arithmetic average) are effective.

Method A always performs the same or better than Method A$'$. This performance shows that the inference of the honesty of rating values using the chain relation improves the performance. This is also true between the Method A$''$ and PeerTrust PSM.

When the number of the malicious peers is not more than 50 (10%), the arithmetic average performs better than PeerTrust PSM. On the other hand, when the number of the malicious peers is not more than 200 (40%), Method A performs better than Method A$''$ in which trust values are never computed by arithmetic average. Furthermore, although Method A$'$ never uses a chain relation for inference of the reliability of rating values, it performs better

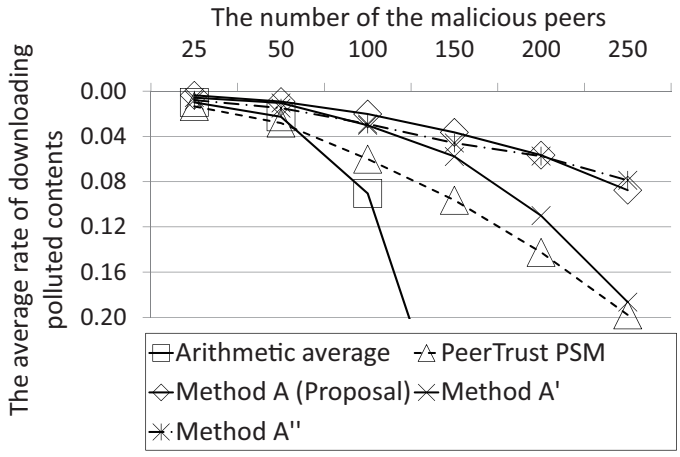


Figure 2.4: The performance according to the number of malicious peers.

than Method A″ when the number of the malicious peers is not more than 50 (10%). These results show that arithmetic average is effective when the ratio of dishonest raters is low.

Method A performs less than Method A″ when the number of the malicious peers is 250 (50%). When the number of the malicious peers is 250, the ratio of dishonest raters is 60%; the numbers of the malicious and the confusing peers are 250 (50%) and 50 (10%), respectively. Since Method A″ never computes trust values by the arithmetic average, the result shows that Method A is deteriorated by arithmetic average. However, Method A is more useful than the Method A″ since the ratio of dishonest raters seldom exceeds 50% in practice as described in Section 2.1.

2.7 Concluding Remarks

In this chapter, only the simple peer behavior is assumed. In addition, a method of trust value computation, which achieves high performance regardless of the ratio of dishonest raters, is proposed. The proposed method infers the reliability of other raters more widely than the existent methods using a chain relation among peers that have a common ratee. Furthermore, if all raters of responding peers cannot be inferred, the proposed method computes trust values by an arithmetic average to achieve high performance especially when the ratio of dishonest raters is low. As a result of the simulation, the proposed method performs better than any other existent methods regardless of the ratio of dishonest raters.

The proposed method needs to trace a chain of peers with common ratees that connects two peers. The proposed method requires more memory, computing, and communication resources than existent methods since existent methods trace only other raters that have a common ratee at most. Efficient and secure implementation of proposed method in P2P systems is beyond the scope of this dissertation, but it is significant future research.

Chapter 3

Trust Value Computation for Probabilistic Peers

3.1 Introduction

In a content sharing system, peers issue queries to request content and to download content from responding peers. A reputation system that is composed of peers and a database of rating values. A reputation system that computes trust values is applied to avoid polluted contents. There can be malicious peers that rate dishonestly to degrade the reliability of trust values. Furthermore, the ratio of dishonest raters or the probabilities providing polluted contents and rating dishonestly largely fluctuate because of the open and anonymous nature of P2P systems. Therefore, a method that computes reliable trust values in such situation is necessary to avoid polluted contents.

In Chapter 2, only the simple peer behavior, where peers always rate either honestly or dishonestly and provide either honest or polluted contents, is considered. However, the probabilistic behavior is investigated in previous works [10, 11, 19, 23–27, 29, 31, 33] because it confuses the reputation system by the probabilistic behavior [10, 11]. The probabilistically peer rates honestly or dishonestly with probability and provides honest or polluted contents with probability. Since the proposed method in Chapter 2 estimates the peer's trustworthiness through a binary decision, it is not sufficient for the probabilistic peer model. Hence, a method that performs well for the model is necessary.

In this chapter, a method for computing reliable trust values regardless of the ratio of dishonest raters or the probability of rating dishonestly is proposed. Since the probabilistic peer behavior is assumed, the proposed method computes the trust value of a peer to be the probability that the

Table 3.1: Probabilities for the probabilistic peer model

$\Pr_d(j)$	The probability that peer j rates dishonestly
$\Pr_p(x)$	The probability that peer x provides polluted contents

peer provides honest contents. First, the proposed method computes the weight between two peers to be the probability that the rating values by the two peers agree. Even when there are no common ratees between the two peers, the weight for the two peers is computed using a chain of peers that have common ratees. Furthermore, if all of the computed trust values are undefined or ineffective in selecting peers to avoid polluted contents despite computed weights, the undefined trust values are recomputed by an arithmetic average by excluding rating values by obvious dishonest raters. To evaluate the effectiveness of the proposed method, a simulation of content sharing is conducted with existent methods.

The remainder of this chapter is as follows. Section 3.2 describes the probabilistic peer model. Sections 3.3 and 3.4 provide a proposal for the probabilistic peer model and its evaluation, respectively. Finally, Section 3.5 discusses the conclusions of this chapter.

3.2 The Probabilistic Peer Model

In this chapter, the probabilistic peer behavior is assumed. The probabilistically behaving peer rates dishonestly with probability and provides polluted contents with probability. Notations of the probabilities are shown in Table 3.1. Probabilistic ratings and providing of polluted contents are defined as follows.

As described in Section 2.2.1, peer $j \in P$ updates information inf_j by the result of perception (i.e., whether downloaded contents are honest or polluted) to compute rt_{jx} ($x \in P$) at each download. $\Pr_d(j)$ is the probability of peer j. Peer j updates inf_j incorrectly with probability $\Pr_d(j)$ i.e., registers the opposite to the result of perception while it updates correctly with probability $(1 - \Pr_d(j))$. We define the probability of a dishonest rating by peer j as $\Pr_d(j)$.

The other aspect of the peer is as a content provider. As described in Section 2.2.1, when peer $x \in P$ is requested to provide content $c \in C$, peer x provides content that is either content c or polluted contents. $\Pr_p(x)$ is the probability of peer x. Peer x provides polluted contents with probability

$\Pr_p(x)$ when requested while it provides honest contents with probability $(1 - \Pr_p(x))$. The probability of providing polluted contents by peer x is $\Pr_p(x)$.

3.3 Proposed Method for the Probabilistic Peer Model

3.3.1 Design of the Method

A method of trust value computation is proposed to achieve high performance regardless of the probabilities of rating dishonestly and providing polluted contents in addition to the ratio of the number of malicious peers. As described in Section 2.3.1, peer i computes trust values tv_{ix} $(x \in P)$ from weights $\mathbf{w}_i$ and rating values M_{rt} by the function f'', as in Equation (2.1).

Since the probabilistic peer behavior is assumed as described in Section 3.2, the function g which calculates rating values had better be defined so that it calculates rt_{jx} to be the probability that ratee peer x provides honest contents observed by rater peer j.

Computation of weights for rating values in related works [10, 11, 19, 23–27, 29–33] is provided in Section 2.3.2. As detailed in the section, weights by raters' similarity [11, 23, 26] are not affected by the ratio of dishonest raters. High performance is preferred regardless of the ratio of dishonest raters. However, in the existent methods, since the raters' similarity for two peers is set to zero when there is no common ratee between the two peers, the trust values of all responding peers may be undefined. Subsequently, computing by the arithmetic average [19, 32] is the most reliable when the ratio of dishonest raters are low while computing using raters' similarity [11, 23, 26] is the most reliable when the ratio of dishonest raters is high.

Our method uses the raters' similarity as the weight of rating value since the similarity is not deteriorated by the ratio of dishonest raters. Our method also computes weight to be the probability that rating values by the two peers agree since we assume the probabilistic peer model. Moreover, to improve in computing the weight of two peers that have no common ratee, a chain of peers that have common ratees is used, and trust values are computed with the weights (see Section 3.3.2). However, a chain does not always exist and subsequently, weights are not always computed properly. Hence, in our method, computed trust values of peers in P_c are judged whether they are effective or not in selecting the peer that provides honest contents from responding peers (see Section 3.3.3). If at least one trust value is judged to be effective, the peer with the highest trust value among P_c is selected. On

the other hand, if judged to be ineffective, the trust values of peers, whose trust values are "undefined" in former computation because all the weights are zero, are recomputed by excluding the rating values by obviously dishonest raters (see Section 3.3.4). Note that there is an apprehension that the performance is not so high when the ratio of dishonest raters is more than 1/3 since a peer may be selected based on the trust value by the arithmetic average, which occurs especially when peers join the system. However, selection based on the trust value by the arithmetic average will not be considerably frequent and the performance will be not so deteriorated especially when the ratio of dishonest raters is no more than 50% because of the widely computing weights that use a chain of peers that have common ratees.

3.3.2 Computing Raters' Similarity and Trust Values

To compute the trust value tv_{ix} of peer x in P_c, peer i first computes the weight wt_{ij} of rating values by peer $j \in P$ by the function f'. Next, peer i computes tv_{ix} from the weight wt_{ij} and the rating value for peer x by peer j, rt_{jx}, by the function f''.

As described in Section 2.2.1, the rating value rt_{jx} is managed and made public as the value rt_{init} when peer j registers no rating values for peer x. In our method, rt_{init} is assumed to be out of the domain range of the function g and it can be seen whether peer j registers the rating value for peer x by examining whether the value of rt_{jx} is rt_{init} or not. Let $R_x = \{j \in P : rt_{jx} \neq rt_{\text{init}}\}$ be the set of peers, called raters for peer x, that register rating value for peer x and $CR_{ij} = \{x \in P : i, j \in R_x\}$ be the set of common ratees of peer i and j. Furthermore, let sim_{ij} be the raters' similarity between peers i and j computed from rating values for common ratees CR_{ij}. We call this similarity the direct raters' similarity because this similarity is computed for two peers using only existing common ratees between the two peers. sim_{ij} is computed as the probability that the rating values by peers i and j agree. Since computing the probability precisely is difficult, it is assumed that the probability is inferred from M_{rt} by a certain method. Note that the example of sim_{ij} is given by Equation (3.6) in Section 3.4, while specific definition of sim_{ij} is not given here. Note that, when $CR_{ij} = \emptyset$, sim_{ij} is set to 0.

Below are shown the computation of the weight wt_{ij} and the definition of the function f'' that calculates the trust values.

Computing the Weight wt_{ij}

Generally, if there are sufficient common ratees between peers i and j, sim_{ij} can be computed to be near the probability that rating values for a peer

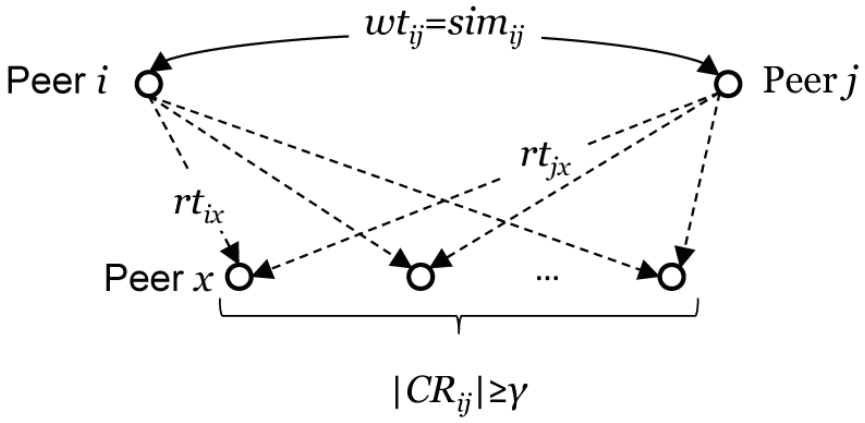


Figure 3.1: Computation of wt_{ij} when $|CR_{ij}| \geq \gamma$.

by peers i and j agree. Hence, the weight wt_{ij} is set to the direct raters' similarity sim_{ij} as

$$wt_{ij} = sim_{ij} \quad (|CR_{ij}| \geq \gamma) \tag{3.1}$$

when the number of common ratees is γ or more where γ is the parameter. Computation of wt_{ij} when $|CR_{ij}| \geq \gamma$ is depicted in Figure 3.1.

On the other hand, when the number of common ratees $|CR_{ij}|$ is less than γ, the weight wt_{ij} is computed using a chain of peers that have common ratees. Let $p = (p_0, \cdots, p_d)$ be the sequence connecting peers i and j where $p_0 = i, p_d = j$, and $p_k \neq p_l$ if $k \neq l$, and any adjoining peers have at least one common ratee. Naturally, the direct raters' similarity between the adjoining peers can be computed. The probability that the rating values by two peers for the other peer agree is depends only on the two peers and the common ratee since the probabilistic peer behavior is assumed. Hence, the direct raters' similarities on the sequence p are independent. Therefore, when there are enough common ratees between the adjoining peers, the product of the direct raters' similarities on the sequence p is the lower bound of the probability that rating values for a common ratee between peers i and j agree. The value of product is seemed to be a proper inference of sim_{ij} with enough common ratees between peers i and j. The value of wt_{ij} is computed by the product of the direct raters' similarities on the sequence where any adjoining peers have μ or more common ratees. Note that the length of the sequence is limited by $d_{\max}$ and only the sequence whose length is the minimum among possible sequences. If there are multiple sequences whose length are the minimum, the weight is computed by the average of the products of the direct raters' similarities on the sequences. The weight is

computed using sequences whose length are the minimum because the longer the sequence, the farther from the value of sim_{ij} with enough common ratees the value of the products, since the value of direct similarity is between 0 and 1. The product of the direct raters' similarities on the sequences connecting peers i and j, sim'_{ij} is computed as

$$sim'_{ij} = \begin{cases} \dfrac{\sum_{p \in SP_{ij}^{\mu,d\max}} \prod_{e=0}^{d_{ij}-1} sim_{p_e p_{e+1}}}{|SP_{ij}^{\mu,d\max}|} & (SP_{ij}^{\mu,d\max} \neq \emptyset) \\ 0 & (\text{otherwise}) \end{cases} ,$$

where $SP_{ij}^{\mu,d\max}$ is the set of sequence $p = (p_0 = i, \cdots, p_{d_{ij}} = j)$ connecting peers i and j where any adjoining peers have μ or more common ratees. The length of sequences in p is d_{ij} ($\leq d_{\max}$) that is the minimum among possible sequences. sim'_{ij} is called the secondhand similarity between peers i and j because this similarity is computed not using common ratees between peers i and j, which is opposite to the direct raters' similarity. When the number of common ratees between peers i and j is less than γ, wt_{ij} is computed as

$$wt_{ij} = sim'_{ij} \quad (|CR_{ij}| < \gamma) \ . \tag{3.2}$$

Figure 3.2 depicts an example of computing wt_{ij} ($|CR_{ij}| < \gamma$) when $d_{ij} = 2$, $|SP_{ij}^{\mu,d\max}| = 3$.

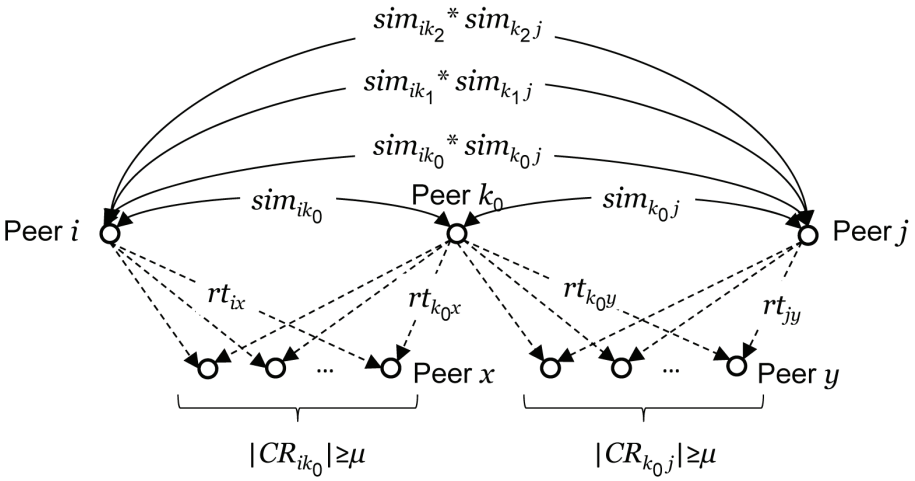
Computing the Trust Value tv_{ix}

The trust value tv_{ix} is computed by the weighted average of rt_{jx} ($j \in R_x$) where the weight is wt_{ij}. Hence, the function f'' is defined as

$$tv_{ix} = f''(\mathbf{w}_i, M_{rt})_x = \begin{cases} \displaystyle\sum_{j \in R_x} \frac{wt_{ij}}{\sum_{j' \in R_x} wt_{ij'}} rt_{jx} & (\textstyle\sum_{j' \in R_x} wt_{ij'} \neq 0) \\ \text{undefined} & (\text{otherwise}) \end{cases} . \tag{3.3}$$

3.3.3 Judging the Effectiveness of Trust Values

The trust values tv_{ix} ($x \in P_c$) computed by Equation (3.3) are judged effective in selecting the peer that provides honest contents from P_c. When tv_{ix} is not "undefined", tv_{ix} can be considered to represent the probability that peer x provides honest contents. Hence, when the maximum trust values of peers in P_c are not sufficiently high, it is not good to select the peer whose trust value is the highest. Therefore, when there is at least one peer $x' \in P_c$ such as $tv_{ix'} > \lambda$ for the threshold λ, the trust values are judged to be effective. Otherwise, they are judged to be ineffective.



$$wt_{ij} = sim'_{ij} = \frac{sim_{ik_0} * sim_{k_0 j} + sim_{ik_1} * sim_{k_1 j} + sim_{ik_2} * sim_{k_2 j}}{3}$$

Figure 3.2: An example of computing wt_{ij} ($|CR_{ij}| < \gamma$) when $d_{ij} = 2$, $|SP_{ij}^{\mu, d_{\max}}| = 3$.

3.3.4 Selecting a Peer

Based on the computed trust values and the result of judgment as in Sections 3.3.2 and 3.3.3 respectively, one peer in P_c is selected to request content to be provided. If the trust values are judged to be effective, the peer whose trust value is the highest is selected.

On the other hand, if judged not to be effective, all of the trust values of peers in P_c computed by Equation (3.3) are either less than λ or "undefined". Since peers whose trust values are less than λ are considered to be untrustworthy, the method tries anew to compute the trust values "undefined" by Equation (3.3) and select a peer. Let $\hat{P}_c \subseteq P_c$ be the set of peers in P_c whose trust values are "undefined". If $\hat{P}_c = \emptyset$, one peer in P_c with the highest trust value is selected.

If $\hat{P}_c \neq \emptyset$, the trust values of peer $x \in \hat{P}_c$ are computed by the arithmetic average of the rating values except for the values by peers that are obviously dishonest raters. Concretely, peer j appears to be a dishonest rater if $CR_{ij} \neq \emptyset$ and $wt_{ij} = 0$. Hence, the rating value rt_{jx} by such peer j is excluded when computing the trust value of peer x by the arithmetic average. Let $R_x^i = R_x \setminus \{j \in R_x : CR_{ij} \neq \emptyset, sim_{ij} = 0\}$. The trust value of peer $x \in \hat{P}_c$,

tv_{ix} is recomputed as

$$tv_{ix} = \begin{cases} \frac{\sum_{j \in R_x^i} rt_{jx}}{|R_x^i|} & (R_x^i \neq \emptyset) \\ \text{undefined} & (\text{otherwise}) \end{cases} . \tag{3.4}$$

Before selecting a peer, the trust values computed by Equation (3.4) are judged effective in selecting the peer that provides honest contents. The judgment is done with the threshold λ as well as in Section 3.3.3. If the trust values are judged to be effective, one peer with the highest trust value in $\hat{P}_c$ is selected. If judged to be ineffective and there is at least one peer whose trust value is "undefined", one peer is selected randomly from "undefined" peers. If judged to be ineffective but there are no peers whose trust values are "undefined", one peer with the highest trust value in P_c is selected.

3.4 Evaluation of Proposed Method

3.4.1 Simulation Settings

The evaluation of the proposed method is conducted. The settings of the simulation are nearly the same as the settings in Chapter 2. The settings of the simulation in Chapter 2 are described in Section 2.6.1. In this section, only the differences from the settings in Chapter 2 are described.

Compared Methods

The proposed method in this chapter is compared with the arithmetic average [32] and PeerTrust PSM [11], which is the same as in Chapter 2. Furthermore, in this chapter, the variances of the proposed method are also compared to evaluate the elements in the proposed method. To evaluate the effectiveness of the secondhand similarity, the trust value computation, which does not use the weight by the secondhand similarity, is used. In the computation, the weight wt_{ij} is defined as

$$wt_{ij} = \begin{cases} sim_{ij} & (|CR_{ij}| \geq \gamma) \\ 0 & (\text{otherwise}) \end{cases} .$$

In addition, the effectiveness of trust value computation by the average in the proposed method is also evaluated. Hence, in this computation, if trust values computed by Equation (3.3) are judged to be ineffective, one peer is randomly selected from the peers whose trust values are "undefined". In the rest of this chapter, the proposed method, the trust value computation that

Table 3.2: Definition of probabilistic malicious peer.

Model	Providing	Rating
Probabilistic malicious peer	polluted contents with probability $\Pr_p(M)$	Rating dishonestly with probability $\Pr_d(M)$

does not use the secondhand similarity, and the trust value computation that does not use the average computation are called "Method B", "Method B′", and "Method B″", respectively.

Assumed Peer Models

In the simulation of this chapter, the honest peer, the confusing peer, and the probabilistic malicious peer are assumed. The malicious peer in Chapter 2 is replaced by the probabilistic malicious peer. A probabilistic malicious peer provides polluted contents and rates dishonestly with probabilities $\Pr_p(M)$ and $\Pr_d(M)$ respectively, as shown in Table 3.2.

The Definition of the Rating Value

In the simulation, the rating value rt_{jx} and the direct raters' similarity sim_{ij} are defined as follows. rt_{jx} is defined to compute the probability that peer x provides honest contents based on the observation by peer j. Hence, rt_{jx} is defined as

$$rt_{jx} = \begin{cases} \dfrac{hd_{jx}}{hd_{jx} + pd_{jx}} & (j \in R_x) \\ rt_{\text{init}} & (\text{otherwise}) \end{cases}, \tag{3.5}$$

where hd_{jx} and pd_{jx} are the stored numbers of downloads of honest and polluted contents from peer x, respectively. These numbers are stored at inf_j. Note that the stored numbers may differ from the result of perception by peer j when downloading, if peer j is not a honest rater i.e., $\Pr_d(j) < 1.0$.

The Definition of Direct Raters' Similarity

For peers $i, j \in P$, sim_{ij} is defined as

$$sim_{ij} = \begin{cases} \dfrac{|\{x : |rt_{ix} - rt_{jx}| \leq \epsilon, x \in CR_{ij}\}|}{|CR_{ij}|} & (CR_{ij} \neq \emptyset) \\ 0 & (\text{otherwise}) \end{cases}, \tag{3.6}$$

Table 3.3: The default values of parameters in the simulation.

# of all peers $\|P\|$	500
# of all honest contents $\|C\|$	(# of the honest peers)*10
ϵ	0.1
λ	0.5
$d_{\max}$	6
# of experiments over which results are averaged	5

where ϵ is the threshold that determines whether two rating value by the two peers are close or not.

When peer i is an honest rater (i.e. the probability of rating honestly is 1) and the probability that peer x provides polluted contents with probability $\Pr_p(x)$, rt_{ix} converges to $(1-\Pr_p(x))$ provided that peer i downloads content from peer x enough times. When peer j is also an honest rater, rt_{jx} also converges to $(1-\Pr_p(x))$. Therefore, ϵ should be set to 0. However, ϵ is set to a little more than 0 because there is an error in the computation of rating values in practice.

Simulations with the various values of ϵ to investigate the appropriate value are conducted. As a result of the simulations, the performance is almost equal when $\epsilon \in [0:1)$ and it is higher than when $\epsilon = 1$. It is confirmed that the number that a peer downloads from another peer is practically at most 1 during content sharing of the simulations. Therefore, while $\epsilon \in [0:1)$, the value of rt_{jx} is practically either 0 or 1 from Equation (3.5); hence, the trueness of $|rt_{ix} - rt_{jx}| \leq \epsilon$ $(x \in CR_{ij})$ and the value of sim_{ij} are seldom varied. This is why the performance is almost equal when $\epsilon \in [0:1)$. On the other hand, when $\epsilon = 1$, $|rt_{ix} - rt_{jx}| \leq \epsilon$ $(x \in CR_{ij})$ is always true since $0 \leq rt_{jx} \leq 1$. Therefore, the performance with $\epsilon = 1$ is lower than the performance with $\epsilon \in [0:1)$ because the direct raters' similarity to another peer with $\epsilon = 1$ cannot be other than 1 although the peer may be a dishonest rater. In Section 3.4.2, only the results of the simulations with $\epsilon = 0.1$ are shown.

Other Settings

Table 3.3 shows the default values of parameters in the simulations. The default value of λ is set to 0.5. The relation of the performance and the value of λ is discussed at the end of Section 3.4.2.

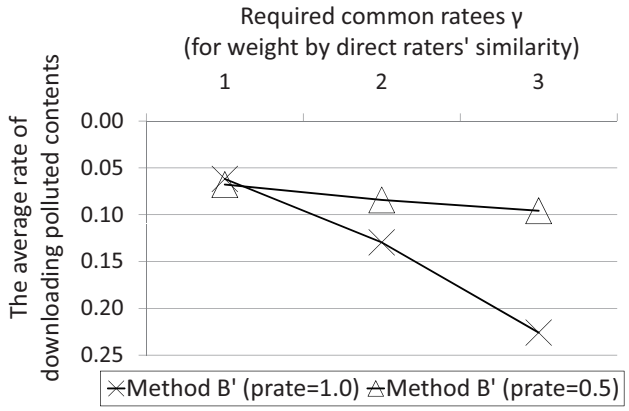


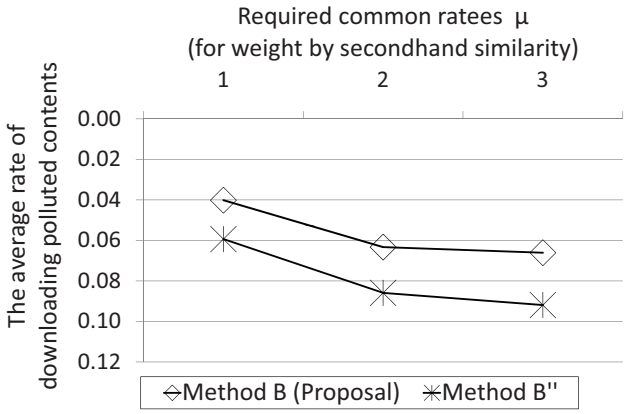
Figure 3.3: The performance of Method B′ according to γ.

3.4.2 Results

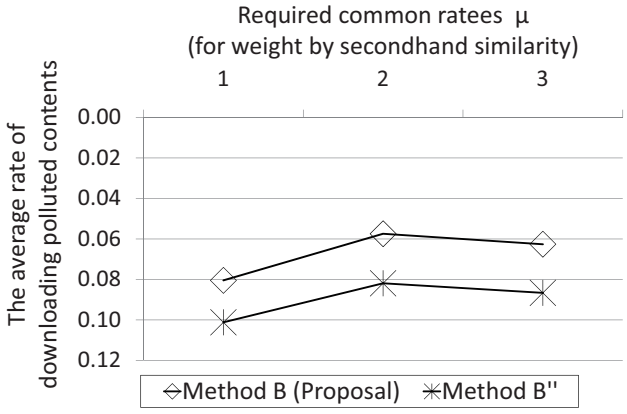
The Performance according to γ and μ

γ and μ are the parameters for computing the weight wt_{ij} $(i, j \in P)$. The greater the value of γ or μ, the more reliable the computed weight. Increasing the value of γ or μ, which is the necessary number of common ratees, implies that content sharing (i.e., peers' querying for and providing content, and rating) has to be made more and more. Therefore, the greater the γ, the more the frequencies that a peer is selected by the trust values computed by the secondhand similarity in Method B (the proposed method) and Method B″, and by the arithmetic average in Method B′. The greater the μ, the more the frequencies that a peer is selected randomly or by the trust values computed by the arithmetic average in Method B and Method B″. Hence, increasing the value of γ or μ does not always improve the performance of the system. Simulations with various values of γ or μ are conducted to investigate their appropriate values.

First, influence for Method B′ by the parameter γ is investigated. Figure 3.3 shows the average rate of downloading polluted contents according to the value of $\gamma = 1, 2, 3$. The numbers of the probabilistic malicious and the confusing peers are set to 150 (30%) and 50 (10%) of 500, respectively. The two lines are for the probabilities that the probabilistic malicious peers provide polluted contents $\Pr_p(M) = 1, 0.5$, respectively. The average rates decrease according to γ regardless of $\Pr_p(M)$. This decrease is because increasing γ makes Method B′ depend more on the trust values computed by the arithmetic average. Below, γ is always set to 1 for Method B′.



(a) $\Pr_p(M) = 1.0$.



(b) $\Pr_p(M) = 0.5$.

Figure 3.4: The performance according to μ.

Next, influence for Method B (the proposed method) and Method B$''$ by the parameter μ is investigated. Figure 3.4 shows the average rate of downloading polluted contents according to the value of $\mu = 1, 2, 3$. The numbers of the probabilistic malicious and the confusing peers are set to 150 (30%) and 50 (10%) of 500, respectively. Figures 3.4(a) and 3.4(b) are for the probabilities that the malicious peers provide polluted contents $\mathrm{Pr}_p(M) = 1.0, 0.5$, respectively. The same simulations with $\gamma = 1, 2, 3$ are also conducted. As a result, the methods perform better with $\gamma = 1$ than the other values of γ. Hence, only the results for $\gamma = 1$ are shown here.

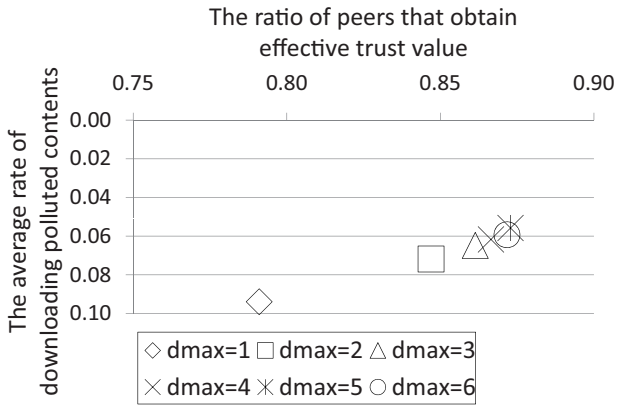
When $\mathrm{Pr}_p(M) = 1$ (Figure 3.4(a)), the performance of the methods decreases while μ increases. This is because the greater the value of μ, the more the performance largely depends on peer selection by randomness or the trust values computed by the arithmetic average. When $\gamma = 2, 3$, the methods with $\mu = 1$ perform better than the other values of μ as well as when $\gamma = 1$.

On the other hand, when $\mathrm{Pr}_p(M) = 0.5$ (Figure 3.4(b)), both of the methods with $\mu = 2$ perform better than $\mu = 1, 3$. When $\mu = 1$, only one common ratee is not enough to compute the reliable direct raters' similarities and the weights since there are peers whose probability that provide polluted contents $\mathrm{Pr}_p(M) = 0.5$, which is between 0 and 1. When $\mu = 3$, the performance of the methods decreases compared to when $\mu = 2$ because the performance largely depends on peer selection by randomness or the trust values by the arithmetic average.

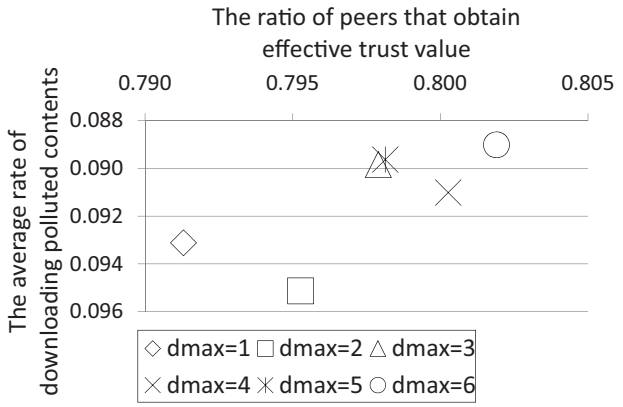
The same simulations by varying the number of the probabilistic malicious peers, or the values of $\mathrm{Pr}_p(M)$ and $\mathrm{Pr}_d(M)$ are conducted. The difference in of the performances between when $\mathrm{Pr}_p(M)$ and $\mathrm{Pr}_d(M)$ are 1 or 0 and when they are between 1 and 0 tends to be the same as the above results. For a probabilistic malicious peer, it appears to be easy to make the probability of providing polluted contents be between 0 and 1. Furthermore, as shown in Figure 3.4, when $\mathrm{Pr}_p(M)$ varies from 1 to 0.5, the declinations of the performances of Method B and Method B$''$ with $(\gamma, \mu) = (1, 2)$ are smaller than those with the other values of (γ, μ). Therefore, the values of the parameters should be set to $(\gamma, \mu) = (1, 2)$. Below, (γ, μ) is always set to $(1, 2)$ for Method B and Method B$''$ if there is no annotation.

The Influence of $d_{\max}$

Computed trust values are judged effective (see Section 3.3.3) frequently and the performance appears to improve more by increasing the value of $d_{\max}$ which is the maximum length of the sequence of peers with at least one common ratee. Hence, we investigate the relevance of Method B$''$ between



(a) $(\gamma, \mu) = (1, 1)$.



(b) $(\gamma, \mu) = (1, 2)$.

Figure 3.5: The average ratio of peers that obtain an effective trust value against the average rates of downloading polluted contents.

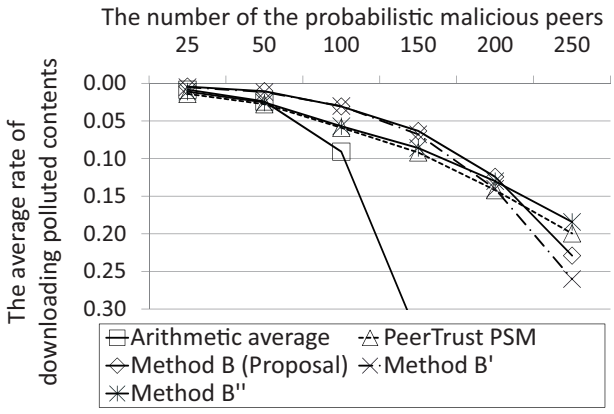


Figure 3.6: The performance according to the number of the probabilistic malicious peers.

the ratio of peers that obtain at least one trust value judged to be effective and the average rates of downloading polluted contents with a varying $d_{\max}$. The ratio of peers that obtain at least one trust value judged to be effective is shown as the results by averaged over an initial 20 cycles because the ratio of peers that obtain at least one trust value judged to be effective converges to 1 until the 20th cycle over all the values of $d_{\max}$ in the conducted simulations. In addition, the average rate of downloading polluted contents is measured after the 20th cycle has finished. The parameters (γ, μ) are set to $(1, 1)$ or $(1, 2)$. The numbers of the honest, the probabilistic malicious, and the confusing peers are 300 (60%), 150 (30%), and 50 (10%), respectively.

Figure 3.5 shows the average ratio of peers that obtain an effective trust value against the average rates of downloading polluted contents for some values of $d_{\max}$. The results for $(\gamma, \mu) = (1, 1)$ and $(1, 2)$ are shown in Figures 3.5(a) and 3.5(b) respectively. By increasing $d_{\max}$, the average ratio of peers that obtain effective trust value increases while the average rates of downloading polluted contents decreases. Therefore, the performance of the method improves by increasing $d_{\max}$.

The Influence of the Ratio of the Probabilistic Malicious Peers

To investigate the influence of the ratios of the dishonest raters and providers, simulations are conducted by varying the number of the probabilistic malicious peers. Figure 3.6 shows the average rates of downloading polluted contents against the number of the probabilistic malicious peers. The number of the probabilistic malicious peers is between 25 (5%) and 250 (50%) of

500 and the number of the confusing peers is set to 50 (10%) of 500. The probabilities that the probabilistic malicious peer provides polluted contents and rates dishonestly, $\Pr_p(M)$ and $\Pr_d(M)$ respectively, are set to 1.0.

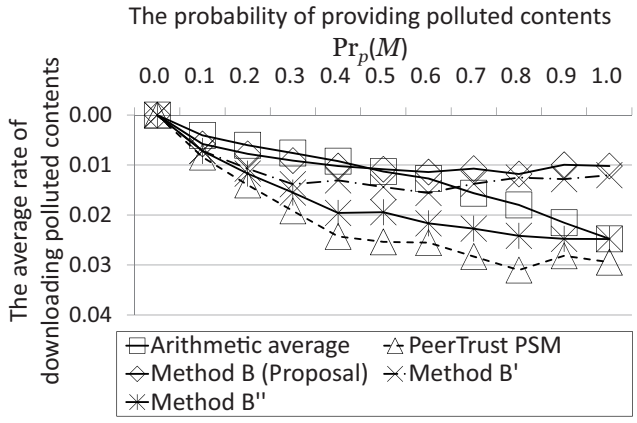
As shown in Figure 3.6, Method B (the proposed method) performs best when the number of the probabilistic malicious peers is not more than 200 (40%) while it performs less than Method B″ and PeerTrust PSM when the number of the probabilistic malicious peers is 250 (50%) or more. This result is because Method B uses trust values computed by the arithmetic average, which is readily deteriorated by the ratio of dishonest raters. Method B′, which does not use weights computed by secondhand similarities, also uses trust values computed by the arithmetic average; however, Method B performs better. Method B performs better than Method B' implies that the secondhand similarity is responsible for deterioration because of the increased ratio of the probabilistic malicious peers.

The Influence of the Probability of Providing Polluted Contents

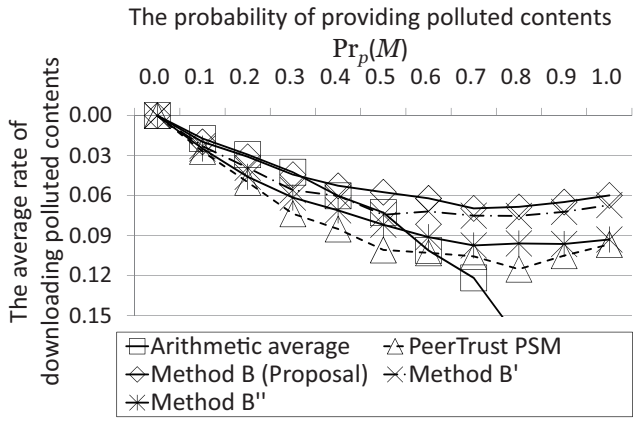
To investigate the influence of the probability of providing polluted contents, a simulation with the varied values of $\Pr_p(M)$, which is the probability that the probabilistic malicious peers provide polluted contents, is conducted. Figures 3.7(a) and 3.7(b) show the average rates of downloading polluted contents against the values of $\Pr_p(M)$ where the numbers of the probabilistic malicious peers are 50 (10%) and 150 (30%) of 500, respectively. The number of confusing peers is 50 (10%) and the value of $\Pr_d(M)$ is always set to 1.0. Note that the settings for $\Pr_p(M) = 1.0$ in Figures 3.7(a) and 3.7(b) are quite equal to the numbers of the probabilistic malicious peers, which are 50 and 150 in Figure 3.6, respectively.

When the number of the probabilistic malicious peers is 50 (10%) as shown in Figure 3.7(a), Method B (the proposed method) performs the best for $\Pr_p(M) \geq 0.5$, while it performs next to the arithmetic average for $\Pr_p(M) \leq 0.4$. The improvement of Method B to Method B′ implies that the improvement is attributed to the secondhand raters' similarity. Furthermore, the improvement of Method B and Method B′ to Method B″ and PeerTrust PSM implies that using trust values by the arithmetic average is effective.

On the other hand, when the number of the probabilistic malicious peers is 150 (30%) as shown in Figure 3.7(b), Method B performs the best for all the values of $\Pr_p(M)$, while the arithmetic average performs as well for $\Pr_p(M) \leq 0.3$. The result implies that improvement of Method B is due to both the secondhand similarity and the arithmetic average as in the case where the number of the probabilistic malicious peers is 50 (10%).



(a) # of the probabilistic malicious peers: 50.



(b) # of the probabilistic malicious peers: 150.

Figure 3.7: The performance according to the probability of providing polluted contents.

The Influence of the Probability of Dishonest Rating

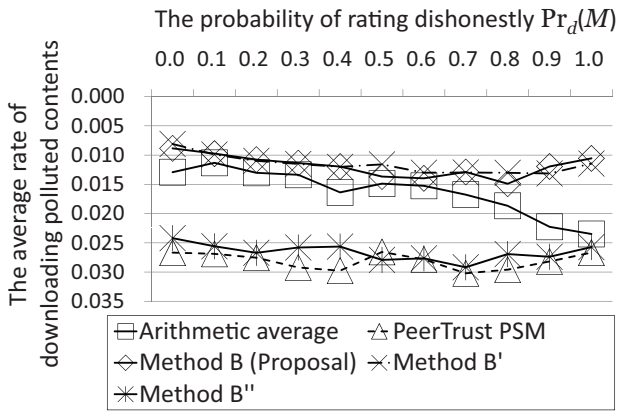
To investigate the influence of the probability of dishonest rating, a simulation with varied values of $\Pr_d(M)$, which is the probability that the probabilistic malicious peers rates dishonestly, is conducted. Figures 3.8(a) and 3.8(b) show the average rates of downloading polluted contents against the values of $\Pr_d(M)$ where the numbers of the probabilistic malicious peers are 50 (10%) and 150 (30%) of 500, respectively. The number of the confusing peers is 50 (10%) and the value of $\Pr_p(M)$ is always set to 1.0. Note that the settings for $\Pr_d(M) = 1.0$ in Figures 3.8(a) and 3.8(b) are quite equal to the numbers of the probabilistic malicious peers, which are 50 and 150 in Figure 3.6, respectively.

When the number of the probabilistic malicious peers is 50 (10%) as shown in Figure 3.8(a), Method B (the proposed method) performs the best for all the values of $\Pr_d(M)$ as well as Method B$'$. This implies that trust values computed by the arithmetic average are especially effective since these methods perform better than Method B$''$.

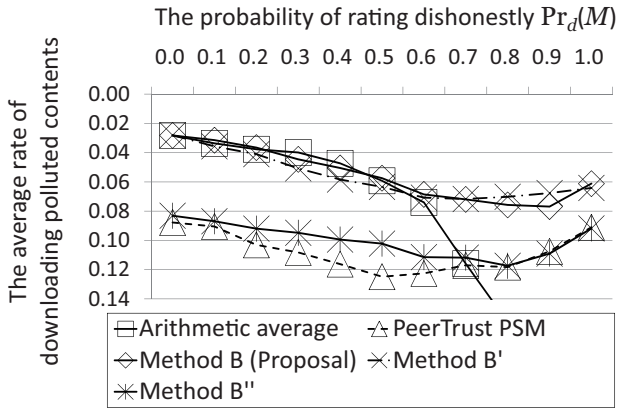
On the other hand, when the number of the probabilistic malicious peers is 150 (30%) as shown in Figure 3.8(b), Method B performs the best for all the values of $\Pr_d(M)$, while the arithmetic average performs as well as Method B for $\Pr_d(M) \leq 0.5$, and Method B$'$ performs as well as Method B for $\Pr_d(M) \geq 0.6$. Trust values computed by the arithmetic average are especially effective when the number of the probabilistic malicious peers is 50 (10%).

The Influence of λ

λ is the parameter of the proposed method for judging the effectiveness of trust values computed by the weighted or arithmetic average as described in Sections 3.3.3 and 3.3.4. If the trust value is λ or more, it is judged to be effective, otherwise, ineffective. If at least one trust value of a peer in responding peers is the appropriate value of λ depends on which is greater, the probability of providing honest contents by the peer whose trust value is the highest in responding peers, or the average probability of providing honest contents among the peers whose trust values are "undefined". This dependence is because one peer whose trust value is the highest in responding peers is selected when at least one trust value is judged to be effective while one peer is randomly selected from peers whose trust values are "undefined" when all trust values are judged to be ineffective. Hence, if the probability that the peer with the highest trust value provides honest contents is greater, λ should be set to a smaller value to select the peer. On the other hand, if the



(a) # of the probabilistic malicious peers: 50.



(b) # of the probabilistic malicious peers: 150.

Figure 3.8: The performance according to the probability of rating dishonestly.

average probability of providing honest contents among the peers whose trust values are "undefined" is greater, λ should be set to a greater value to select randomly from the peers with the trust values "undefined". However, the peer that is about to select cannot know in practice which of the probabilities is greater. Therefore, it appears to be appropriate that λ is set to 0.5, which is the middle of 0.0 and 1.0 since trust values are between 0.0 and 1.0.

To investigate the influence of λ experimentally, the same simulation of Method B (the proposed method), B$'$, and B$''$ with λ set to -1 (< 0) or some values in $[0 : 1]$ are conducted. When $\lambda = -1$ (< 0), a peer is selected by trust values computed with secondhand or direct raters' similarity except in the case whare all of the trust values are "undefined" by Equation (3.3) since all of the trust values which are not "undefined" are judged to be effective. Therefore, Method B with $\lambda = -1$ performs as well as Method B$''$. When $\lambda = 1$, a peer is selected randomly from a peer whose trust values are "undefined" except for the case where all of the trust values are not "undefined" since all of the trust values which are not "undefined" are judged to be ineffective. Method B with $\lambda = 1$ performs better than it does with $\lambda = 0.5$ when the ratio of dishonest raters is more than 50% while Method B with $\lambda = 1$ performs less than Method B with $\lambda = 0.5$ when the ratio of dishonest raters is not more than 50%. When the ratio of dishonest raters is low (i.e., not more than 50%), because trust values by the arithmetic average are sufficiently reliable but, a peer is selected randomly by Method B with $\lambda = 1$, Method B performs less than it does with $\lambda = 0.5$. Contrarily, when the ratio of dishonest raters is high (i.e., more than 50%), trust values by the arithmetic average are not reliable; hence, Method B with $\lambda = 1$ in which peers are frequently randomly selected performs better than it does with $\lambda = 0.5$.

Furthermore, some values of λ in $[0 : 1)$ are investigated. As a result, while $0 \leq \lambda < 0.8$, Method B performs constantly. When $0.8 \leq \lambda < 1$ and the ratio of dishonest raters is high, Method B performs slightly better than it does with $0 \leq \lambda < 0.8$ and as well as it does with $\lambda = 1$. When $0.8 \leq \lambda < 1$, but the ratio of dishonest raters is low, Method B performs as well as it does with $0 \leq \lambda < 0.8$. Only the results with $\lambda = 0.5$ are shown since Method B performs almost constantly when $0 \leq \lambda < 1$.

3.4.3 Discussions

As described in Section 3.4.2, Method B (the proposed method) performs almost the best. The result of the simulation that varies the number of the probabilistic malicious peers shown in Figure 3.6 implies that Method B is virtually effective regardless of the ratio of the number of dishonest raters.

Method B performs the best when the ratio of dishonest raters is not more than 50%, which is composed of the 200 probabilistic malicious peers with $\Pr_d(M) = 1$ and the 50 confusing peers of 500 peers. On the other hand, Method B performs less than Method B$''$ and PeerTrust PSM when the ratio is 60%, which is composed of the 250 probabilistic malicious peers with $\Pr_d(M) = 1$ and the 50 confusing peers of 500 peers. However, Method B is considered virtually effective regardless the ratio of dishonest raters because the ratio over 50% is impractical as described in Section 2.1.

Another simulation that varies the probability $\Pr_p(M)$ so that the 50 (10%) of probabilistic malicious peers rate dishonestly is shown in Figure 3.7(a), where Method B and Method B$'$ perform less than the arithmetic average when $\Pr_p(M) \leq 0.4$. Since the ratio of dishonest raters is low, when $\Pr_p(M) \leq 0.4$, the methods can perform better than the arithmetic average by modifying the judgement of the effectiveness of trust values described in Section 3.3.3 so that larger part of the trust values are computed by the arithmetic average as in Equation (3.4). However, such modification seems to cause degradation of the performance when the ratio of dishonest raters is high. Because of this apprehension and the degradation of Method B to the arithmetic average when $\Pr_p(M) \leq 0.4$ is smaller than the improvement to $\Pr_p(M) \geq 0.5$ as shown in Figure 3.7(a), the judgment in Section 3.3.3 is better than the modification.

3.5 Concluding Remarks

In this chapter, a method for computing reliable trust values regardless of the ratio of dishonest raters or the probability of rating dishonestly is proposed. The proposed method computes weights for rating values in computing trust values not only by direct raters' similarity, but also secondhand raters' similarity even with peers without common ratees, using a chain of peers that have common ratees. Furthermore, when trust values are computed with secondhand raters' similarity as the weight but all of the computed trust values are "undefined" or less than the threshold, trust values that are "undefined" are anew computed by excluding rating values by obvious dishonest raters. As a result of the conducted simulations, the proposed method performs as well as the arithmetic average when the ratio of dishonest raters is less than 15% and performs better than that of existent methods using the raters' similarity when the ratio is less than 50%.

In this chapter, the probabilistically behaving peer was considered a practical peer behavior. Changing behavior by time is another major practical model [36]. Making our proposal to handle a time model is another avenue

for future research.

Chapter 4

Overlay Construction for Prevention of Polluted Contents

4.1 Introduction

In a P2P content sharing system, search efficiency is a significant performance metric in addition to preventability of polluted contents. When a peer downloads content, it selects a source of downloading content from peers responding query. Even if a peer can precisely estimate the trustworthiness of the responding peers, the peer initiating query cannot obtain its desired content when there is no trustworthy peer in the responding peers. Hence, a trustworthy peer should be frequently included in responding peers for both an efficient search and prevention of polluted contents. Prior to downloading, searching for desired contents is processed. A searching should be processed efficiently.

In the system, peers form an overlay which is a logical network to search for content. Regardless of methods of forwarding query messages or structures of an overlay, a query message for searching for content travels a path on an overlay. If a peer receiving query message responds, the query message never travels in most searching methods. A peer natrurally responds to a received query when the peer holds the contents matching the query or the peer is about to provide polluted contents although the peer does not hold the contents matching the query. Therefore, the line-up of responding peers and the number of links through which a query message travels largely depend on the position of peers around the peer initiating query on an overlay. This situation implies that the topology of the overlay is significant for search

efficiency and preventability of polluted contents.

For each peer, the topology of the overlay is such that other peers with high probabilities of providing honest contents (PHC) and holding desired contents (HDC) are neighbors or within a short distance, which is preferable for efficient search and prevention of polluted contents. In practice, even among honest peers, PHC over the honest peers is varied because they provide polluted contents mistakenly. HDC is also varied because held contents reflect each peer's individual interests. Therefore, unfortunately, peers where both PHC and HDC are high are only a small part of all peers; hence, all of the peers may not achieve optimum performance. A reasonable solution is that peers with high PHCs are privileged to achieve high performance, i.e., to take peers that both have PHCs and HDCs as high as their neighbors. The solution rewards peers with high PHCs.

In previous works, various methods of topology adaptation are proposed [12, 17–22]. Topology adaptation is a process of constructing an overlay by each peer's individual selection of its neighbors. In topology adaptation, neighbors are selected based on the results of an evaluation of other peers. In existent methods, other peers are evaluated by either PHC or HDC. In methods evaluated by PHC, peers whose PHCs are high are grouped on an overlay; however, the peers are not clustered into peers whose HDCs are mutually high. On the other hand, in methods evaluated by HDC, all peers are clustered into peers whose HDCs are mutually high; nonetheless, peers whose PHCs are high are not grouped. Hence, existent methods do not achieve a reasonable performance.

In this chapter, a method of topology adaptation is proposed to achieve both efficient search and preventability of polluted contents especially so that peers with high PHC are rewarded. In the proposed method, other peers are evaluated by both PHC and HDC while PHC is preceded. Peers with high PHCs are grouped on an overlay by preceding PHCs. Furthermore, peers with high PHCs are clustered into peers whose HDCs are mutually high by evaluating HDCs. To show the effectiveness of our method, simulation of content sharing with topology adaptation is conducted by comparing some of the existent methods.

The remainder of this chapter is as follows. First, Section 4.2 describes content sharing on an onverlay and overlay construction by topology adaptation. Section 4.3 describes the design of existent and proposed methods. Section 4.4 describes the proposed method, and Section 4.5 evaluates this method. Finally, Section 4.6 discusses the conclusions of this chapter.

Table 4.1: Significant parameters.

$B_i \subset P$	The set of neighbors of peer $i \in P$ on overlay
$\tau^i_{\max}$	The allowed number of neighbors of peer i
$R_i \subset P$	The set of candidates for connection request of peer $i \in P$
$H_c \subseteq P$	The set of peers that hold content c

4.2 Content Sharing with Topology Adaptation

In this chapter, a content sharing system on an unstructured overlay constructed by topology adaptation is considered. An unstructured overlay is a logical network formed by peers where peers may take any other peer as a neighbor. On the contrary, a structured overlay is a logical network where neighboring peer is restricted by the peer's ID, e.g., Chord [13], CAN [14], Tapestry [15], and Pastry [16]. Unstructured overlay is considered rather than a structured overlay because the structured overlay is unsuitable for constructing an overlay that satifies an efficient search and preventability of polluted contents due to the limitation on neighboring peer by peer's ID, which is irrelevant to the peer's behavior. Topology adaptation is a process that constructs an overlay by each peer's individual selection of its neighbors. Concrete processes are proposed in [12, 17–22]. In this section, the content sharing process on an unstructured overlay and a framework of topology adaptation are described.

Table 4.1, in addition to Tables 2.1 and 3.1, shows additional significant parameters. The set of all peers on an overlay is denoted by P and the set of all honest contents shared by peers in P is denoted by C. $\Pr_p(x)$ is the probability that peer x provides polluted contents. It is assumed that the probability affects not only the providing contents, but also the response to the query message (see Section 4.2.1). $B_i \subset P$ is the set of neighbors of peer $i \in P$ on an overlay. $\tau^i_{\max}$ is the allowed number of neighbors, which is described in Section 4.2.1. $R_i \subset P$ is the set of candidates given in the connection request process (see Section 4.2.2) executed by peer $i \in P$. Since peers in R_i may be connected in the future, $B_i \cap R_i = \emptyset$. H_c is the set of peers that hold content c.

4.2.1 Content Sharing on an Unstructured Overlay

It is assumed that the number of neighbors on an overlay is limited, i.e., $|B_i|$ $(i \in P)$ has its own maximum $\tau^i_{\max}$. The limitation depends on the capacities of the peer, e.g., on the capacities of the physical network and computation, etc. An unstructured overlay is used when peers search for content. When a peer searches its desired content, the peer first sends a query message to its neighbors. The receiving peer responds to the peer initiating the query, forwards the query message to its neighbors, or throws such messages away. Finally, the peer initiating query selects one peer with the highest trust value in the responding peers and downloads content from the selected peer. Trust values are computed by the reputation system described in Section 2.2.1. Note that the peer downloads nothing if no peer responds.

Forwarding Query Message

In this chapter, query messages are forwarded on an overlay by flooding [37]. Note that there are various methods of query forwarding [38–42] other than flooding. Flooding is based on the bredth first search (i.e., query messages are forwarded to all neighbors), while other methods narrow links through which query messages are forwarded probabilistically or deterministically. Hence, the number of forwarded query messages at flooding is upper bound of other forwarding methods. Therefore, only assuming flooding is sufficient for evaluation of overlay with respect to preventability of polluted contents and search efficiency.

In flooding, peer $i \in P$ initiates a search by sending query messages for its desired content $c \in C$ to all its neighbors B_i. When peer $j \in P$ receives a query message, peer j determines whether to throw it away immediately or not. Immediately throwing away a message avoids duplicate processing of query messages. Each query message has a message ID embedded by the initiating peer i. The message ID is generated by a random number generator to avoid collision of the message IDs among all forwarded messages on an overlay. When receiving a query message, peer j immediately throws the message away if its message ID is one of the message IDs which peer j has processed.

When not immediately throwing away the received query message, peer j determines whether to respond to the received query message or not. When responding, peer j provides peer i about information which is necessary to communicate, such as the peer's ID, IP address, etc. This dissertation assumes that peer j responds to peer i directly. The received query message is never forwarded.

When not responding, peer j either forwards the query messages further or throws them away. The query message is forward further if the number of links through which the message is forwarded is less than the threshold. Otherwise, the message is thrown away. The threshold is the parameter called TTL (Time-To-Live), which limits the number of links through which the query message travels. A peer i initiating query embeds the variable ttl whose value is the threshold TTL. When not responding, peer j first subtracts 1 from the value of variable ttl embedded in the message. Subsequently, if the value of ttl is more than 0, peer j copies the query message and forwards it to its neighbors except for the peer that had sent the query message. Otherwise, if the value of ttl is 0, peer j throws away the message.

Responding Query

As mentioned, the peer determines whether to respond to the received query or not. A peer responds only when it has contents matching the query or intends to provide polluted contents. Let a peer search for content $c \in C$. When peer $j \in H_c$ receives the query message, j always responds to the initiating peer directly. When peer $k \notin H_c$ receives, peer k responds to the initiating peer with probability $\Pr_p(k)$ to provide polluted contents.

As the processes of flooding, it is assumed that a responding peer directly responds to a peer initiating the query. In a variant of flooding, a responding message by the responidng peer for the initiating peer is fowarded reversely along the path though which the query message has traveled to the responding peer. In the variant of flooding, a malicious peer that aims to disseminate polluted contents on the path may intercept the reverse forwarding of the responding message [9]. However, since the malicious peer has received the query message corresponding the responding message before, it is sufficient for dissemination of polluted contents that the malicious peer responds to provide polluted contents when recieving the query message. Hence, assuming that a responding peer directly responds to a initiating peer is sufficient.

Note that an Eclipse attack [5] is not coped with in this dissertation. A peer k with $\Pr_p(k) > 0$ intercepts forwarding query messages but it is only an instance of an attacker on an Eclipse attack. Because an Eclipse attack aims to intercept query forwading, a peer that does not provide polluted contents but intercepts query forwarding is also an attacker. However, in this dissertation, a peer where interception of query forwarding is influenced by only the probability of providing polluted contents is assumed.

Assumed Distributions of Peers' Parameters

In this chapter, the probabilities of providing honest contents and holding another peer's desired contents are considered the parameters of a peer. The probability of providing honest contents is the reverse of providing polluted contents, i.e., $(1 - \mathrm{Pr}_p(x))$ for peer x, which affects query responding as described immediately above. Below, the probabilities that a peer provides honest contents and that a peer holds another peer's desired contents are called PHC and HDC, respectively. The probability that peer i holds peer x's desired contents is denoted by $\mathrm{Pr}_h(i, x)$.

It is assumed that PHC and HDC over honest peers are distributed normally. The background of the distributions is as follows:

- PHC
  If peer x is honest, $\mathrm{Pr}_p(x)$ is ideally 0. However, in practice, $\mathrm{Pr}_p(x)$ is rarely 0 even if peer x is honest because a peer may provide polluted contents mistakenly. Therefore, it is assumed that $\mathrm{Pr}_p(x)$ over honest peers x is distributed normally. PHC of peer x (i.e., $(1 - \mathrm{Pr}_p(x))$) are naturally distributed normally.

- HDC
  Peers are interested in the subset of existing content in practical P2P content sharing systems [43]. In addition, it is observed that the number of peers that have content is proportional to Zipf distribution of the popularity ranks of all content [8]. Therefore, content is categorized and the number of peers holding or desiring contents in a category is proportional to Zipf distribution of popularity ranks of all categories [44]. Because the occupied proportion of the distribution by the highest rank is large and the proportion for each rank declines quickly with ranks in Zipf distribution, HDC over all peers is distributed normally. The normal distribution of HDCs over all peers is observed by simulation with the model described in [44].

Note that PHC and HDC of a peer are assumed to be independent because the frequency of providing polluted contents mistakenly and the peer's interests appear to be independent in nature.

4.2.2 Topology Adaptation

In the topology adaptation, each peer on the overlay selects a member of its neighbors. Subsequently, the overlay is constructed. Each peer individually evaluates other peers, and connects or disconnects them based on the result

of evaluation [12, 17–22]. The topology adaptation is composed of the processes for connection request and connection acceptance [12, 17–22]. Peer $i \in P$ executes the process of connection request and requests a peer in R_i to connect on an overlay. When peer $x \in P$ receives a connection request from peer i, peer x executes the process for connection acceptance. If peer x accepts the connection request as a result of the process, peers i and x have link on an overlay. Note that a peer may disconnect from its existing neighbors to keep the number of neighbors not more than its limitation when executing the processes.

The Connection Request

A peer requests another peer to connect by executing the connection request process. In the process, the peer selects a peer from given candidates R_i to request to connect. Next, peer i executes the process. The function $Req_i(R_i, B_i, inf2_i(R_i \cup B_i))$ evaluates peers in R_i or B_i, and selects a peer in R_i as a result. $inf2_i(R_i \cup B_i)$ is the information on each peer in R_i and B_i for evaluation. Peer i may select no peer when no peer in R_i is satisfactory.

The Connection Acceptance

The connection acceptance process is executed by a peer when the peer is requested to connect. In the process, the peer judges whether to accept or reject the request. Next, peer x executes the process which is triggered by the request from peer i. The function $Acp_x(i, B_x, inf2_x(\{i\} \cup B_x))$ evaluates peer i and its neighbors B_x, and determines whether peer x accepts the request or not. If peer x accepts, peer i and x are connected. In particular, if peer x accepts when the number of current neighbors $|B_x|$ is at its limitation $\tau^x_{\max}$, peer x disconnects an existing neighbor whose rank is the lowest in B_x to keep the number of neighbors not more than $\tau^x_{\max}$.

4.3 Design of Topology Adaptation

Search efficiency and preventability of polluted contents are important performance measures of the system, and largely depend on the topology of the overlay. From Section 4.2.2, the key to design has been how to evaluate candidates of neighbors in the connection request, or how to evaluate a peer requesting a connection in the connection acceptance, i.e., in the design of the functions Req_i and Acp_x $(i, x \in P)$. We consider this design to achieve both prevention of polluted contents and efficient search.

4.3.1 Relation of the Performance and Overlay

Performance Metric

Search efficieny and preventability of polluted contents are used as performance measures of contents sharing. The number of forwarded query messages during searching and the highest PHC in responding peers quantify search efficiency. On the other hand, preventability of polluted contents is also quantified by the highest PHC in responding peers. Highness of the highest PHC in responding peers implies that the peer with the highest PHC has the contents matching the query with high probability because the higher the PHC, the lower the probability of responding without the contents matching the query according to the model of query responding described in Section 4.2.1. In addition, the peer initiating query selects the peer with the highest trust value, which is an inference of PHC in responding peers as described in Section 4.2.1. Hence, by assuming that computed trust values are reliable, the higher the highest PHC in responding peers, the higher the probability that the peer initiating a query obtains its desired contents. Note that the highest PHC is assumed to be 0 when no peer responds.

Relation between the Performance and Topology

The performance largely depends on the topology of the overlay. Since query messages are forwarded through logical links on an overlay as described in Section 4.2.1, which peers respond and the number of links through which query messages travel largely depend on the topology of the overlay. To achieve high performance, there must be a peer with a high probability of providing honest contents in responding peers. Therefore, first of all, query message should arrive at the peer called the 'preferred peer' with high probabilities of providing honest contents and holding desired contents. On an overlay, the preferred peer should be positioned within the distance TTL from the peer initiating a query because of the process of query forwarding described in Section 4.2.1, i.e., flooding. In addition, peers with a low probability of providing honest contents should not be on a path on which query messagea travel between the peer initiating the query and the preferred peer since a peer with low probability interrupts traveling of query messages although it does not have the contents matching the query. Furthermore, for an efficient search, the distance between the peer initiating the query and the preferred peer should be short because the longer the distance, the more the number of forwarded query messages during searching.

Preferred Topology of Overlay

As mentioned in the previous section, to achieve high performance, a peer whose both PHC and HDC are high is preferred to be a neighbor or to be within a short distance, while peers with a low PHC is not preferred to be on the path from a peer initiating a query and the preferred peer. However, since PHCs and HDCs are distributed normally as described in Section 4.2.1, the preferable peer, i.e., the peer whose both PHC and HDC are high, seldom exists. Hence, only a small part of all peers can take preferable peers as neighbors.

It is reasonable that peers whose PHCs are relatively high are privileged to take the preferable peer as neighbors. Therefore, peers whose PHCs are high and HDCs are mutually high should be grouped on overlay.

4.3.2 The Design of Existent Methods

As described immediately above, the topology of an overlay is significant for performance. Therefore, from Section 4.2.1, the key to design of topology adaptation has been the evaluation of other peers in the processes, i.e., the functions Req_i and Acp_x $(i, x \in P)$, which affects the selection of peers to be neighbors. The information for evaluation, $inf2_i(Q)$ $(Q \subset P)$, which is the input of the functions, is an estimation of either PHC, HDC, or both in existent methods [12, 17–22]. How to estimate is also significant. We review the evaluation of other peers and subsequent topologies as well as the estimation of PHC and HDC in existent methods [12, 17–22].

Limitation on the Evaluation in this Dissertation

In INGA [18], RC-APT [12] and AGP [21], evaluation of forwarding query messages is included in $inf2_i(Q)$ besides in the estimation of PHC or HDC. In INGA, other peers are additionally evaluated with respect to frequency of forwarding query messages which yields downloads of desired contents. In RC-APT, the peer whose PHC is low, but that deliberately has malicious peers as neighbors and forwards query messages to them is assumed. In AGP, a peer that is free-rider on query forwarding messages is assumed. Contrary to these methods, in this dissertation for simplicity, these types of peer are not assumed. Hence, evaluation of query forwarding is beyond the scope of this dissertation. Instead, concentration is put on the behavior of providing content, i.e., the HDC and the PHC. Note that evaluation of query forwarding can be easily integrated with an evaluation of HDC or PHC by the weighted sum of the evaluation results as in INGA, RC-APT, and AGP.

Obtaining HDC and PHC

The HDCs for two peers are obtained by reporting about their own content by themselves. In INGA [18] and AGP [21], two peers mutually report to the other about held contents and compute the value called a relevance score or a similarity score, which is the estimation of HDC of the other peer. In GES [17], a peer issues a query only to estimate HDCs of the other peers. When receiving the query message, a peer responds to the initiating peer with information on its held contents. The initiating peer computes the relevance scores which are the estimations of HDC of the responding peers.

The PHC is obtained by the reputation system [19, 21]. In the reputation system, when a peer downloads content from another peer, the downloading (rater) peer stores the rating value for the providing peer (ratee) in the peer that is in the role of storing the rating value. In STEP [19] and AGP [21], a reputation system is run. In STEP [19], the rating values are stored with some peers except the ratee peer of the rating values. A peer who wants to know the PHC of another peer queries on an overlay by flooding to obtain the rating values for the objective peer and computes the trust value from collected rating values. The computed value is the inference of the PHC. In AGP [21], rating values are collected by querying the objective peer. The queried peer answers the set of rater peers itself.

In [12, 20, 22], both the HDC and the PHC are obtained from the history of direct downloading from the objective peer.

Evaluation and Subsequent Topology

Evaluations of other peers in existent methods are reviewed. In existent methods, information for evaluation is either HDC [17, 18], PHC [19], or both [12, 20–22]. The topology of a constructed overlay is varied according to evaluations.

- HDC
  In GES [17] and INGA [18], a malicious peer is not assumed. Peers are preferred to be neighbors only if their HDCs are high. Since PHC is not dealt with in the methods and PHC is varied even over honest peers as described in Section 4.2.1, peers with relatively low PHCs can be positioned on neighbors. Hence, as a result of the overlay construction by the methods, peers are clustered into peers whose HDCs are mutually high. In addition, peers with relatively low PHCs mingle with clusters, which is different from the preferred overlay described in Section 4.3.1. Therefore, the performance may not be optimum.

- PHC
  In STEP [19], search efficiency is not explicitly considered. Peers are preferred to be neighbors only if their PHCs are high. Hence, peers with relatively high PHCs are the core of the overlay, and peers with relatively low PHCs are around the core as a result of overlay construction by the methods. Since HDC is not considered, HDCs of neighbors are not always high for each peer, which differs from the preferred overlay described in Section 4.3.1. Therefore, the performance may not be optimum.

- Both of HDC and PHC
  In AGP [21], [22], APT [20], and RC-APT [12], peers are evaluated by both HDC and PHC. In AGP [21], a peer requests a peer with the highest PHC in given candidates. However, because existing neighbors and requested peers are not compared with respect to PHC in AGP, the preceding PHC in the evaluation is incomplete. In [12, 20, 22], the objective peer is evaluated by direct experience with the peer. The value for evaluation is the number of successful downloads over the number of sending query messages to the objective peer. Although [22] does not assume malicious peers, the value is the probability of providing desired contents, (HDC)$*$(PHC). In [12, 20], the value for the evaluation is the number of downloads of honest contents minus the number of downloads of polluted contents from the object peer. Because the greater the PHC or the HDC, the greater the value, the order of the peers by the value is roughly equal to the order by (HDC) $*$ (PHC). Hence, peers are virtually evaluated by (HDC) $*$ (PHC) in [12, 20]. Since the probability does not represent the PHC directly, prevention of polluted contents is not achieved sufficiently when there are peers such that the value is high, but the PHC is low. These methods [12, 20–22] treat both HDC and PHC; however, they incompletely achieve a reasonable performance.

4.3.3 Design of Topology Adaptation

Obtaining Information for Evaluation

It is assumed that the HDCs of the objective peer are obtained by mutual reporting as INGA [18] and AGP [21]. In this approach, a dishonest peer may report false information about held contents, which is not held to tempt other peers to be neighbors and consequently disseminate polluted contents. However, since our method evaluates other peers, i.e., not only the HDC but also the PHC, such dishonest peer will excluded eventually.

In addition, it is assumed that the PHC is obtained by the reputation system. A peer collects rating values for the objective peer and computes the PHC. If the PHC is computed from only direct experiences as in APT [20], RC-APT [12], and [22], it is less possible to obtain the PHC than in the case where the PHC is computed from not only direct experience, but also other peers' rating values of the objective peer in a reputation system.

Furthermore, it is assumed that rating values are stored at peers that are not the objective peer as in STEP [19]. In AGP [21], rating values of the objective peer (ratee) are obtained by querying the objective peer. In the case of AGP [21], the objective peer can istself report a falsehood to raise the computed PHC dishonestly. Our assumption differs from AGP [21].

In practice, there could be peers that dishonestly register unfair rating values. However, it is assumed that such rating values are excluded by some proper methods when computing the PHC, e.g., the proposed method in Chapter 3.

Evaluation of Other Peers

As mentioned in Section 4.3.1, peers whose PHCs are high and whose HDCs are mutually high should be grouped on an overlay. As described in Section 4.3.2, methods which evaluate peers only by PHC [12, 19, 20] construct an overlay such that the core is peers whose PHCs are high. On the other hand, methods which evaluate peers by only HDC [17, 18] construct an overlay which is clustered into peers whose HDCs are mutually high. Therefore, in topology adaptation, it appears to be appropriate that both PHC and HDC are evaluated while PHC is preceded. The core of the overlay composed of peers with high PHCs will be constructed by precedence of PHC. Furthermore, peers in the core will be clustered into peers with mutually high HDCs by evaluating HDC. Subsequently, the preferred overlay will be constructed. Hence, in our proposal, encountering peers in the topology adaptation are evaluated by both PHC and HDC, but PHC is prior to HDC. In the connection request (i.e., the function Req_i), peers in candidates and existing neighbors are compared for evaluation. In the connection acceptance (i.e., the function Acp_x), the requesting peer and existing neighbors are compared for evaluation.

Following the philosophy of evaluating other peers, when comparing two peers, two peers are ordered by HDC only if PHCs of the two peers are exactly equal; otherwise, the two peers are ordered by PHC. However, since PHCs of peers are distributed as described in 4.2.1, any two PHCs are virtually never equal exactly. Therefore, in our proposal, the PHCs which are in certain tiny scope are regarded as equal and the peers whose PHCs are in the scope are

compared in the order of HDCs.

4.4 Proposal of Topology Adaptation

In this section, our proposal of topology adaptation (i.e., definitions of the functions Req_i and Acp_x) is described. Prior to describing our proposal, assumptions for obtaining information to evaluate other peers is described.

4.4.1 Assumptions for Obtaining the Information

In the processes of the topology adaptation, the HDC of the objective peer is obtained by mutual reporting. The PHC of the objective peer is obtained by the reputation system. Rating values of the objective peers for computing the PHC are stored at some peers except for the objective peer and dishonest rating values, which are excluded by proper method.

- HDC
  Let $\Pr'_h(i,x)$ be the inferred probability that peer x has contents desired by peer i. When obtaining $\Pr'_h(i,x)$ in the processes, peer i and x mutually reports about each peer's held contents. It is assumed that reported information is some kinds of proper description of held contents, e.g., XML description of held documents described in AGP [21], and computed $\Pr'_h(i,x)$ is moderately correct; hence, the error in the inferred HDC is ignored.

- PHC
  Let tv_{ix} be the trust value of peer x computed by peer i, which is the inferred probability that peer x provides honest contents, i.e., the inferred value of $(1-\Pr_p(x))$. tv_{ix} is obtained by the reputation system. It is assumed that the computed trust value tv_{ix} $(i,x \in P)$ is reliable.

Note that $\Pr'_h(i,x)$ and tv_{ix} for peer $x \in Q$ $(Q \subseteq P)$ are included in $inf2_i(Q)$.

4.4.2 Topology Adaptation

Preparation for the Processes

We define some notations to evaluate peers in the processes. In the following, Q $(\subseteq P)$ is the set of peers, i and x are peers in P, and ψ is a real number.

- $topPrh(i,Q) = \{x \in Q : \Pr'_h(i,x) = \max_{y \in Q} \Pr'_h(i,y)\}$.

Require: $x \in P, Q \subseteq P, \psi, inf2_i(\{x\} \cup Q)$
Ensure: true, false
1: **if** $tv_{ix} > \min_{y \in Q} tv_{iy}$ **then**
2: **return** true
3: **else if** $tv_{ix} \in [\min_{y \in Q} tv_{iy} - \psi, \min_{y \in Q} tv_{iy}]$, $\Pr'_h(i,x) > \Pr'_h(i, y \in btm(Q,i))$ **then**
4: **return** true
5: **else**
6: **return** false
7: **end if**

Figure 4.1: The pseudo codes of the function cmp_i.

- $top(Q, i, \psi) = \{x \in Q : tv_{ix} \in [\max_{y \in Q} tv_{iy} - \psi, \max_{y \in Q} tv_{iy}]\}$.
- $btm(Q,i) = \{x \in btm'(Q,i) : \Pr'_h(i,x) = \min_{y \in btm'(Q)} \Pr'_h(i,y)\}$, where $btm'(Q,i) = \{x \in Q : tv_{ix} = \min_{y \in Q} tv_{iy}\}$.

$top(Q, i, \psi)$ is the set of the peers whose trust values are regarded as equal and the highest in the given set Q. As in the evaluation policy described in Section 4.3.3, PHCs within a certain scope are regarded equal. ψ is the width of the scope. $topPrh(i, Q)$ and $top(Q, i, \psi)$ are used in the function Req_i, which requests other peers to connect. The set $btm(Q, i)$ determines the peers evaluated the worst by peer i in the given set Q. Peers in $btm(Q, i)$ are the peers whose HDCs are the lowest in peers whose PHCs are the lowest in Q. $btm(Q, i)$ is used in the function Acp_i when selecting a peer to disconnect.

Furthermore, the function $cmp_i(x, Q, \psi, inf2_i(\{x\} \cup Q))$ $(i, x \in P, Q \subseteq P)$ is defined as shown in Figure 4.1. The function is used when deciding whether to replace an existing neighbor with a new one or not. The function $cmp_i(x, Q, \psi, inf2_i(\{x\} \cup Q))$ determines whether peer x is evaluated better by peer i than the peer evaluated the worst in Q or not, following the evaluation policy described in Section 4.3.3. The peer evaluated the worst in Q has the minimum HDC within peers with the minimum trust value in Q. When peer x is evaluated higher, cmp_i returns "true", otherwise, "false". If the trust value of peer x is higher than that of the worst peer, peer x is evaluated higher as in the lines 1 and 2. Another case of evaluating higher is when the trust value of peer x is regarded as equal by ψ to that of the worst peer and the HDC of peer x is higher than that of the worst peer as in the lines 3 and 4.

In our method, the candidates' set R_i for the connection request is the set of peers that have responded to peer i, but not to neighbors first. R_i includes peers that have provided contents for peer i. Since responding peers appear

Require: $R_i, B_i, inf2_i(R_i \cup B_i)$
Ensure: true, false
1: **if** $|B_i| < \tau^i_{\max}$ **then**
2: $R'_i \leftarrow \{x \in R_i : \Pr'_h(i,x) > 0\}$
3: **else**
4: $R'_i \leftarrow \{x \in R_i : cmp_i(x, B_i, \psi, inf2_i(\{x\} \cup B_i)) = \text{true}\}$
5: **end if**
6: $R''_i \leftarrow top(R'_i, i, \psi)$
7: **while** $R''_i \neq \emptyset$ **do**
8: Request one peer x randomly selected from $topPrh(R''_i, i)$
9: **if** accepted by peer x **then**
10: **if** $|B_i| = \tau^i_{\max}$ **then**
11: Disconnect one peer in $btm(B_i, i)$
12: **end if**
13: **return** true
14: **else**
15: $R''_i \leftarrow R''_i \setminus \{x\}$
16: **end if**
17: **end while**
18: **return** false

Figure 4.2: The pseudo codes of the function Req_i.

to hold contents desired by peer i, taking responding peers as candidates is effective. This is same as [21, 22]. Furthermore, when peer i cannot connect, R_i is replaced by peers randomly selected from all peers in P to find the most preferable peer. In our method, a peer takes candidates at the most half of all peers in P because half seemes to be sufficient in seeking a preferable peer. Note that randomly selected peers are assumed to be given by the bootsrap server of the system.

The Connection Request

Figure 4.2 shows the pseudo codes of the function Req_i which returns "true" if the connection request is accepted, but otherwise returns "false". As mentioned in the process of seeking new neighbors shown in Figure 4.3, peer i seeks new preferable peers as far as possible. Hence, function Req_i is triggered many times until peer i finds a preferable peer that accepts a connection request. The function Req_i is triggered as in Figure 4.3.

Next, peer i executes the function Req_i. Prior to execution, peer i obtains the information $inf2_i(R_i \cup B_i) = \{(tv_{iy}, \Pr'_h(i,y)) : y \in R_i \cup B_i\}$ first. Peer

1: $R_i \leftarrow$ {Peers that have responded to peer i, but are not its neighbors.}
2: $r \leftarrow |R_i|$
3: Obtain $inf_{R_i \cup B_i}$
4: **if** $Req_i(R_i, B_i, inf_{R_i \cup B_i})$ =true **then**
5: **return**
6: **end if**
7: **while** $r < |P|/2$ **do**
8: $R_i \leftarrow$ {Some randomly selected peers}
9: $r \leftarrow r + |R_i|$
10: Obtain $inf_{R_i \cup B_i}$
11: **if** $Req_i(R_i, B_i, inf_{R_i \cup B_i})$ =true **then**
12: **return**
13: **end if**
14: **end while**
15: **return**

Figure 4.3: The pseudo codes of the process for seeking new preferable peer by peer i.

i requests a part of the peers in the candidates R_i. When any request is accepted, Req_i is terminated immediately.

In the function Req_i, first, the candidates R_i are narrowed to R_i', which is the set of peers satisfying the marginal condition. When the current number of neighbors $|B_i|$ is less than its limitation $\tau^i_{\max}$, any peer may be a candidate. However, a peer with no desired contents is insignificant. Hence, R_i' is the set of peers whose HDC is more than 0 as in the line 2. On the other hand, when $|B_i| = \tau^i_{\max}$, only candidates that are evaluated higher than the worst peer in the existing neighbors is taken. Hence, R_i' is the set of peers evaluated higher than the worst peer in existing neighbors B_i by the function cmp_i as in the line 4. Furthermore, R_i' is narrowed to R_i'' which is the set of peers whose trust values are the highest with the equality of trust value by the width ψ as in the line 6.

Next, peer x requests peers in R_i'' in the order of inferred HDCs as in the lines 7–17. Note that if the request is accepted when $|B_i| = \tau^i_{\max}$, peer i disconnects a peer randomly selected in $btm(B_i, i)$, i.e., the peer evaluated the worst in the existing neighbors, to keep the number of neighbors not more than $\tau^i_{\max}$ as in the lines 10 and 11.

Require: $i, B_x, inf2_x(\{i\} \cup B_x)$
Ensure: true, false
1: **if** $|B_x| < \tau^x_{\max}$ **then**
2: **if** $\Pr'_h(x, i) > 0$ **then**
3: **return** true
4: **else**
5: **return** false
6: **end if**
7: **else**
8: **if** $cmp_x(i, B_x, \psi, inf2_x(\{i\} \cup B_x)) =$ true **then**
9: Disconnect one peer in $btm(B_x, x)$
10: **return** true
11: **else**
12: **return** false
13: **end if**
14: **end if**

Figure 4.4: The pseudo codes of the function Acp_x.

The Connection Acceptance

Let peer x be the requested peer and i be the requesting peer. Peer x obtains $inf2_x(\{i\} \cup B_x) = \{(tv_{xy}, \Pr'_h(x, y)) : y \in \{i\} \cup B_x\}$ at first. Next, peer x executes the function Acp_x as shown in Figure 4.4. The function Acp_x returns "true" if accepted, and "false" otherwise.

When the number of neighbors $|B_x|$ is less than its limitation $\tau^x_{\max}$, peer x accepts if $\Pr'_h(x, i)$ is more than 0, which is the marginal requirement as in the line 2. On the other hand, when $|B_x| = \tau^x_{\max}$, peer x accepts if peer i is evaluated higher the peer evaluated lowest in existing neighbors B_x by the function cmp_x as in the line 8. If accepted, one peer in $btm(B_x, x)$ is disconnected as shown in the line 9.

4.5 Evaluation of Proposed Method

In this section, simulations of topology adaptation are conducted. Our method is compared with some existent methods.

4.5.1 Simulation Settings

The simulation repeats a number of cycles. In each cycle, each peer issues a query on an overlay, obtains responding peers, and executes topology adap-

Table 4.2: Simulation settings.

# of peers $\|P\|$	500
# of cycles in one simulation	300
TTL (Time-To-Live) of flooding	4
Distribution of $\tau^i_{\max}$ $(i \in P)$	Zipf
The maximum value of $\tau^i_{\max}$ over all peers	20
The minimum value of $\tau^i_{\max}$ over all peers	3
Initial # of neighbors $\|B_i\|$ $(i \in P)$ on random graph	3
The distribution of PHC over honest peers	Normal distribution with average 0.95 and variance $4.0 * 10^{-4}$
ψ	0.05
# of experiments over which results are averaged	5

tation as described in Section 4.2. Content sharing is perpetual during the simulation. In the following, the evaluation metric, compared methods, assumed peer models, and distributions of PHC and HDC in the simulations are described. Table 4.2 summarizes the setting of simulations.

Evaluation Metric

Following the performance metric described in 4.3.1, methods are evaluated by the highest trust value of responding peers and the number of forwarded query messages during a search. For each cycle and peer, the highest trust values in responding peers and the number of forworaded query messages are observed. Methods are evaluated by each measurement averaged over repetition cycles and by peers.

Compared Methods

As mentioned in Section 4.3.2, previous methods evaluate other peer in topology adaptation by either HDC [17, 18], PHC [19], or both [12, 20–22]. However, details of the methods (e.g, obtaining information to evaluate and candidates for requesting to connect) differ from our proposal. To evaluate our proposal clearly and simply, compared methods are set to be the same as our proposal other than the evaluation of other peers at the process of topology

adaptation. Therefore, compared methods are variants of our proposal with respect to the evaluation of other peers by HDC, PHC, or both. Concretely, for the function Req_i, peer i requests peers in R_i in the order of HDC, PHC, or the value obtained from PHC and HDC. For the function Acp_x, the requesting peer and existing neighbors are compared by the order of the values. As mentioned in Section 4.3.2, peers are evaluated by (PHC)$*$(HDC) in APT [20], RC-APT [12], and [22]. Hence, peers are ordered by (PHC) $*$ (HDC) in the variant for PHC and HDC. Note that because there is no value for ordering peers to substitute for AGP [21], only AGP is directly picked up in our simulation. However, settings on assumptions for obtaining information and candidates of requesting connection are set the same as our proposal. These setting are described in Section 4.4.1 and the beginning of Section 4.4.2. Below, the variants for PHC, HDC, and PHC and HDC are denoted by "sb. PHC", "sb. HDC", and "(PHC)*(HDC)", respectively.

Settings on Topology of Overlay

Distribution of the allowed number of neighbors and the topology of an overlay at beginning of the simulation are set as follows. The settings are summarized in Table 4.2.

The maximun number of observed node degree in real P2P content sharing systems is 20 [45]. Hence, the maximum value of $\tau^i_{\max}$ among all peers is set to 20. The minimum value of $\tau^i_{\max}$ among all peers is set to 3. Furthermore, the distribution of node degree shows power law in practical P2P content sharing systems [46, 47]. Hence, the values of $\tau^i_{\max}$ ($i \in P$) are derived from values between 3 and 20, to be drawn by Zipf distribution, which is a representative probability distribution of power law.

The initial topology of an overlay is a random graph where each peer has some neighbors, which are set randomly. It is assumed that a peer takes some peers as neighbors, and these neighbors are randomly selected peers from existing on-line peers by the bootstrapping server of the content sharing system (e.g., GWebCache[1]) when a peer joins the system.

Peer Models

In the simulations, all peers in P are composed of honest and malicious peers. Honest peers almost always provide honest contents, but occastionally provide polluted contents by mistake. On the other hand, malicious peers deliberately provide polluted contents. It is assumed that malicious peers aim to disseminate polluted contents over honest peers. Hence, each

[1] http://www.gnucleus.com/gwebcache

malicious peer tries to take honest peers as neighbors. Concretely, a malicious peer always requests one peer selected randomly from the honest peers and accepts a connection request from an honest peer. When a malicious peer j disconnects an existent neighbor to keep the number of neighbors not more than its limitation $\tau^j_{\max}$, peer j randomly selects a neighbor to be disconnected.

Distributions of PHC and HDC

As described in Section 4.2.1, the values of PHCs and HDCs over all honest peers are assumed to be distributed normally. For each peer model, there is an additional condition.

An honest peer provides polluted contents by mistake. Hence, the values of $\mathrm{Pr}_p(x)$ for honest peers x are set to be distributed normally, where the average is close to 0.0. On the other hand, a malicious peer deliberately provides polluted contents. Hence, the values of $\mathrm{Pr}_p(y)$ for malicious peers y are set to a value much more than the average over honest peers. Note that the value of PHC for peer x is $(1 - \mathrm{Pr}_p(x))$. In simulations, the average and the variance of $(1 - \mathrm{Pr}_p(x))$ over all honest peers is set to 0.95 and 10^{-4}, respectively, as shown in Table 4.2.

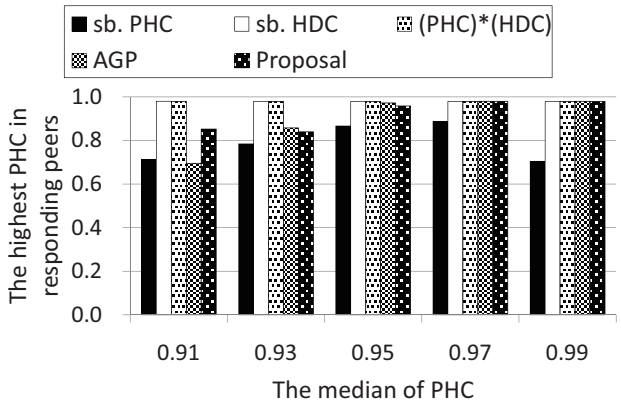
It is natural that the PHC and HDC for a peer are independent. In addition, since held contents reflects the peer's interests, it is natural that HDC is symmetry, i.e., $\mathrm{Pr}_h(i, x) = \mathrm{Pr}_h(x, i)$ $(i, x \in P)$. Hence, the values of $\mathrm{Pr}_h(i, x)$ $(i, x \in P)$ are set to be distributed normally with the condition of the symmetry. Note that the values of HDCs are fixed during the simulation. It assumes that peers that respond a query for each some content are fixed.

4.5.2 Results

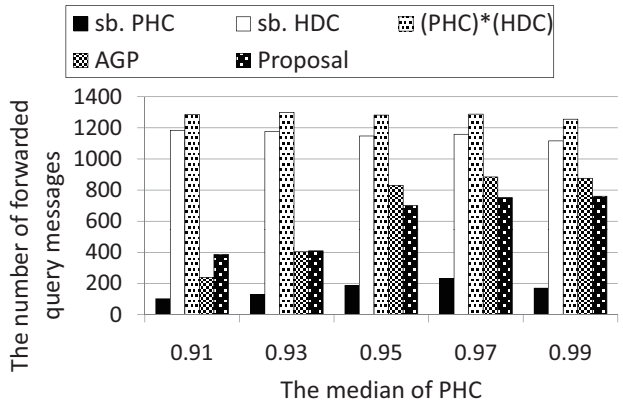
In simulations, performance according to distributions of HDC, and performance against the ratio and the probability of providing polluted contents $\mathrm{Pr}_p(M)$ of malicious peers are investigated.

Performance against the Average of HDC

In practice, distribution of HDC may be varied due to fluctuations of disseminated content. To investigate the performance against distributions of HDC, first, a simulation that varyies the average of HDC is conducted. Figures 4.5 and 4.6 show the highest PHC in responding peers and the number of forwarded query messages during a search versus the medians of ranges of

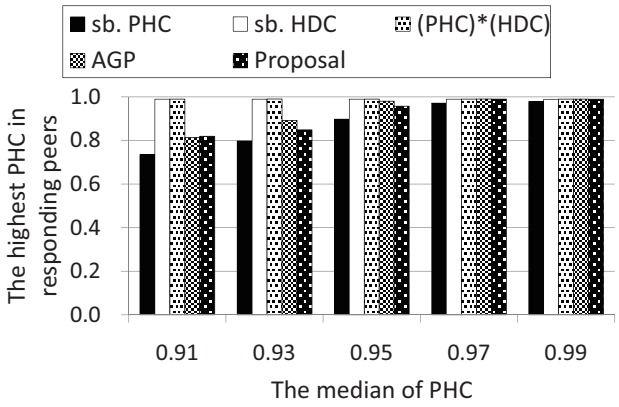


(a) The highest PHC among responding peers.

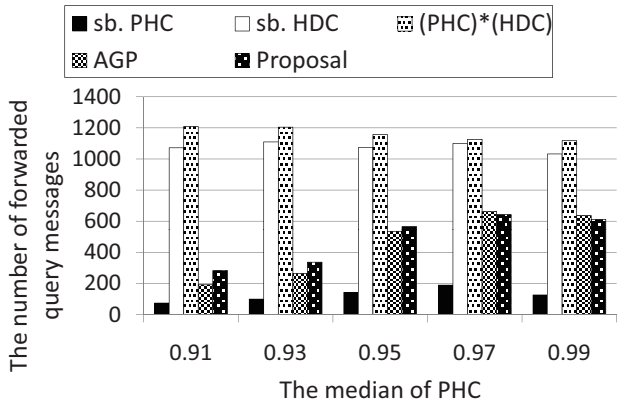


(b) The # of forwarded query messages.

Figure 4.5: Performance of topology adaptation when the average of HDC is 0.01.



(a) The highest PHC among responding peers.



(b) The # of forwarded query messages.

Figure 4.6: Performance of topology adaptation when the average of HDC is 0.1.

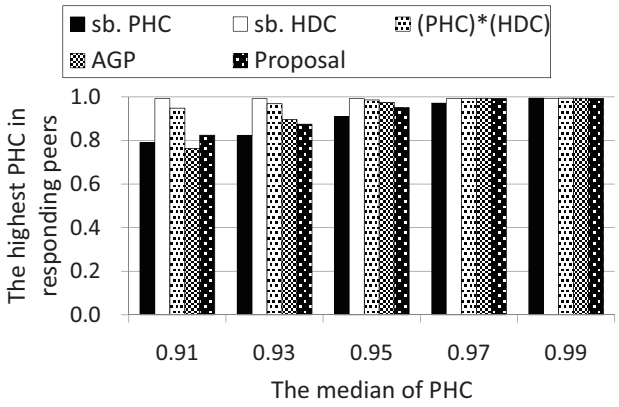
peers' PHC. The result is the average over peers in each range. The averages of HDC are 0.01 for Figure 4.5, and 0.1 for Figure 4.6. The variance is $2.5 * 10^{-5}$ for both. All of peers are set to be honest peers to concentrate on observing the influence by the average of HDC. Because there are only a few peers whose PHCs are less than 0.9 at the distribution of PHC over honest peers, results for these few peers are discarded.

As shown in both figures, the proposed method achieves the value of the highest PHC no less than existent methods in groups of peers with higher PHC, while the numbers of forwarded query messages for the groups are less than other methods except for sb. PHC. sb. PHC reduces the number of forwarded query messages because peers with close PHCs are gathered and cycles frequently appear on an overly. Subsequently, query messages do not travel farther. However, sb. PHC is disadvantageous against the small average of HDC, e.g., when iterest of peers to content is highly divided. As shown in Figure 4.5 where the average of HDC is 0.01, sb. PHC achieves the value of the highest PHC considerably less than other methods. On the contrary, in Figure 4.6 where the average of HDC is 0.1, sb. PHC is not so degraded.

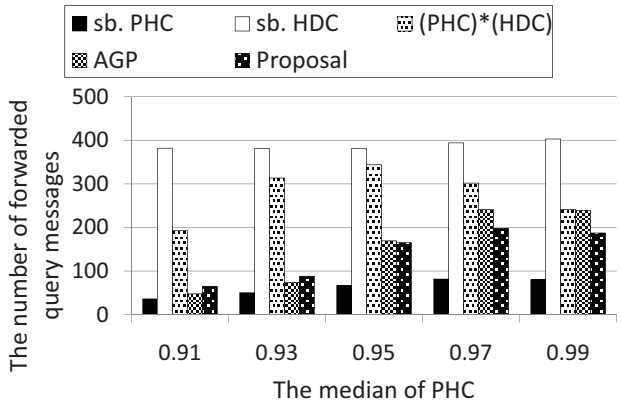
Performance against the Variance of HDC

To investigate the performance against distributions of HDC, a simulation with varying the variance of HDC is conducted. Figures 4.7 and 4.8 show the highest PHC in responding peers and the number of forwarded query messages during a search versus the medians of the ranges of peers' PHC. The setting of the simulation is the same as that for Figures 4.5 and 4.6 except for the distribution of HDC. The variances of HDC are $2.5 * 10^{-5}$ for Figure 4.7, and 0.1 for Figure 4.8, while the average of HDC is 0.5 for both.

Figures 4.7 and 4.8 show that the proposed method achieves the value of the highest PHC no less than existent methods in groups of peers with higher PHC while the numbers of forwarded query messages for the groups are less than in other methods except for sb. PHC. In the methods of sb. HDC, (PHC)*(HDC), AGP, and the proposed method, the number of forwarded query messages decreases by increasing the variance. This decrease is because these methods evaluate other peers in topology adaptation using HDC and peers are clustered into small groups where their HDCs are mutually high when the variance is large. Note that since values of HDC lie in between 0 and 1, and they are assumed to be distributed normally, variance cannot be much greater than in our experiments. Hence, the results shown here reveal the largest change in performance by variance of HDC.

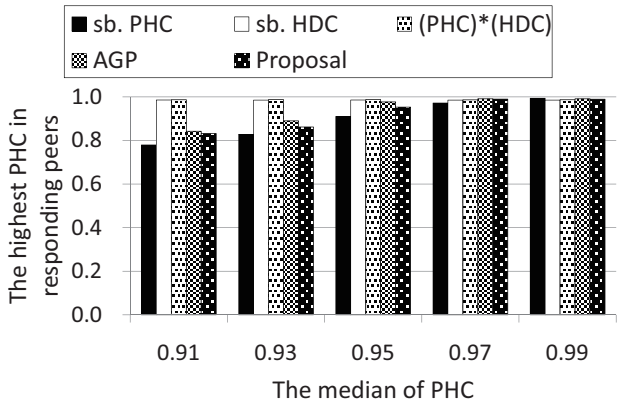


(a) The highest PHC among responding peers.

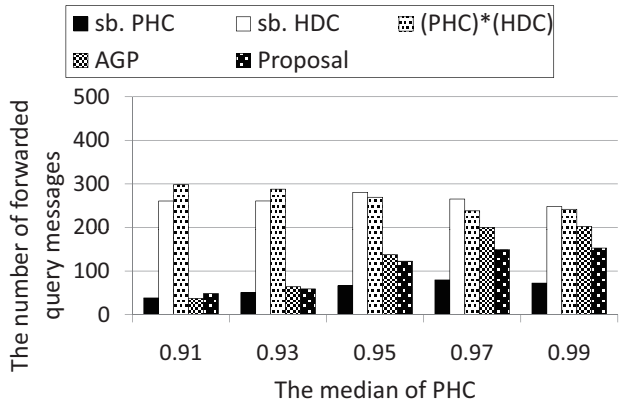


(b) The # of forwarded query messages.

Figure 4.7: Performance of topology adaptation when the variance of HDC is $2.5 * 10^{-5}$.



(a) The highest PHC among responding peers.



(b) The # of forwarded query messages.

Figure 4.8: Performance of topology adaptation when the variance of HDC is 0.1.

Performance against the ratio of Malicious Peers

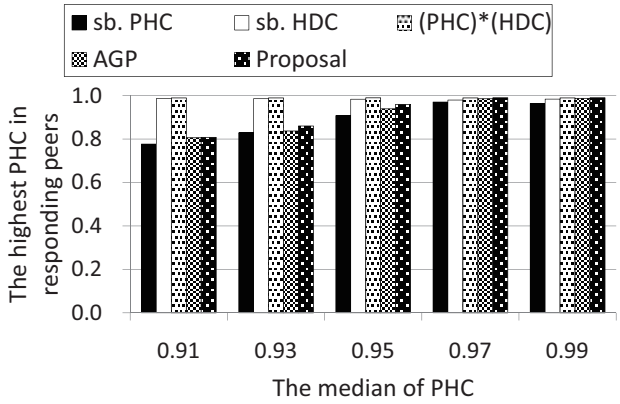
To investigate the influence of the performance for honest peers by malicious peers, simulations by varying the ratio of the number of malicious peers and the probability that malicious peers provide polluted contents are conducted. Figures 4.9 and 4.10 show the highest PHC in responding peers and the number of forwarded query messages during the search versus the medians of ranges of peer's PHC. The numbers of malicious peers are 50 (10%) for Figure 4.9, and 150 (30%) for Figure 4.10. The probability that malicious peers provide polluted contents $\mathrm{Pr}_p(M)$ is set to 1. The distribution of HDC over all peers is set to the normal distribution with the average 0.01 and the variance $2.5 * 10^{-5}$.

Figures 4.9 and 4.10 show that the proposed method achieves the value of the highest PHC no less than existent methods in groups of peers with a higher PHC. When the number of malicious peers is 50 (10%), sb. PHC and AGP reduce the number of forwarded query messages more than the proposed method as shown in Figure 4.9(b); however, these methods achieve lower values of the highest PHC in responding peers than the proposed method. When the number of malicious peers is 150 (30%), sb. HDC reduces the number of forwarded query messages considerably as shown in Figure 4.10(b). This reduction occures because honest peers connect malicious peers since peers are evaluated by only HDC in sb. HDC, and malicious peers always respond since $\mathrm{Pr}_p(M) = 1$.

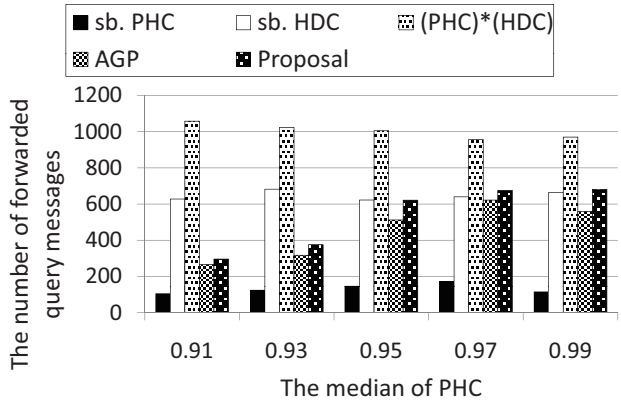
Performance against the probability of providing polluted contents

To investigate the influence of the performance for honest peers by the probability of providing honest contents, simulations by varying the probability that malicious peers provide polluted contents are conducted. Figures 4.11 and 4.12 show the highest PHC in responding peers and the number of forwarded query messages during a search versus the medians of ranges of peer's PHC. The settings on the simulations are the same as those of Figure 4.9 and 4.10 except for the probability that malicious peers provide polluted contents $\mathrm{Pr}_p(M)$. The value of $\mathrm{Pr}_p(M)$ is 0.5 for Figure 4.11 and 0.8 for Figure 4.12, while the number of malicious peers is 150 (30%) for both.

Figures 4.11 and 4.12 show that the proposed method achieves the value of the highest PHC no less than existent methods in groups of peers with a higher PHC. As shown in Figures 4.11(b) and 4.12(b), the numbers of forwarded query messages in the proposed method and AGP are greater than that of other methods. This is because honest peers may connect to malicious peer whose probability of holding honest contents is extremely higher than

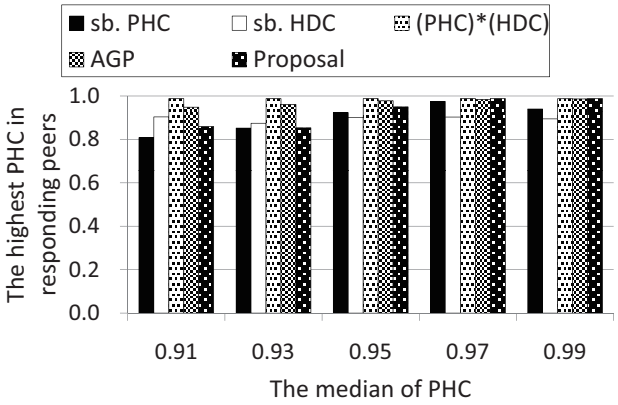


(a) The highest PHC among responding peers.

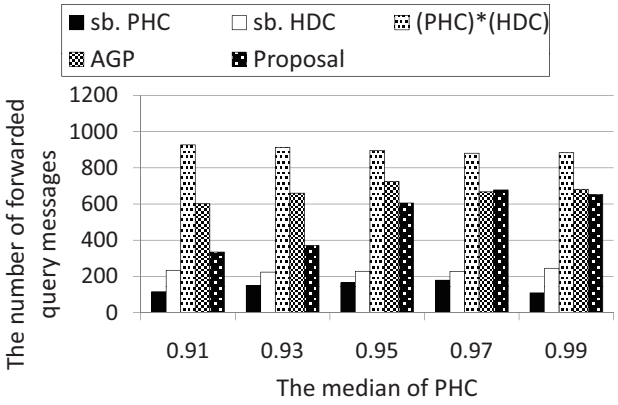


(b) The # of forwarded query messages.

Figure 4.9: Performance of topology adaptation when there are 50 malicious peers.

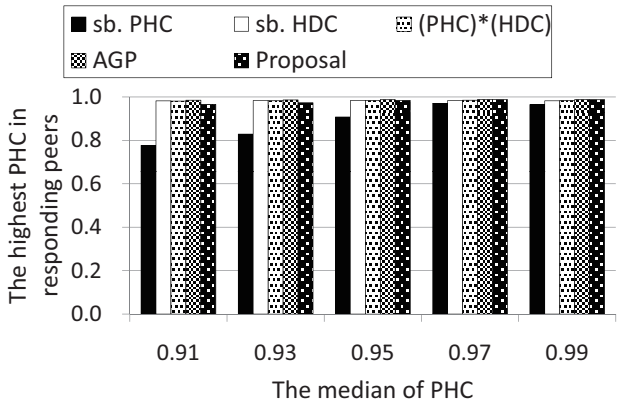


(a) The highest PHC among responding peers.

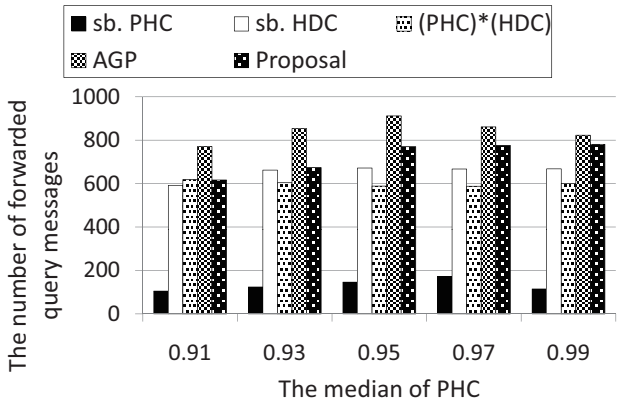


(b) The # of forwarded query messages.

Figure 4.10: Performance of topology adaptation when there are 150 malicious peers.

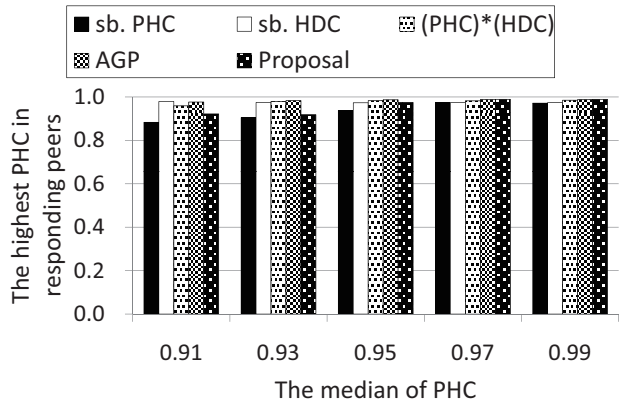


(a) The highest PHC among responding peers.

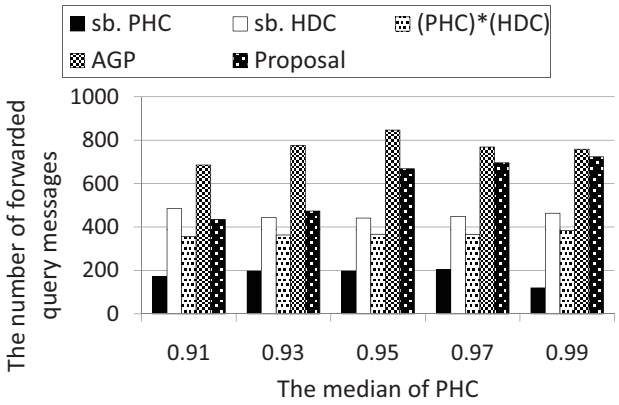


(b) The # of forwarded query messages.

Figure 4.11: Performance of topology adaptation when $\mathrm{Pr}_p(M) = 0.5$.



(a) The highest PHC among responding peers.



(b) The # of forwarded query messages.

Figure 4.12: Performance of topology adaptation when $\Pr_p(M) = 0.8$.

that of honest peers for sb. HDC and (PHC)*(HDC).

4.5.3 Discussion

For simulations assuming only honest peers, the proposed method achieves a reasonable performance regardless of the average or the variance of HDC, i.e., in the proposed method, the value of the highest PHC is no less than existent methods in groups of peers with a higher PHC, while the number of forwarded query messages exceeds other existent methods except for sb. PHC. As shown in Figure 4.5, sb. PHC is disadvantageous when interests of peers to content is highly divided Hence, the proposed method is the most useful when there are only honest peers.

When assuming malicious peers, the number of forwarded query messages of the proposed method partially exceeds that of other methods. This is because, in other methods, peers may connect malicious peers, and malicious peers forward received query messages since their probability of providing polluted contents is much higher than that of honest peers. Hence, the value of the highest PHC in groups of peers with a higher PHC is no less than other methods, and our proposal is the most useful.

4.6 Concluding Remarks

In this chapter, the method of topology adaptation was proposed for both search efficiency and preventability of polluted contents. The proposed method evaluates peers by both the probabilities of holding desired contents and providing honest contents while the probability of providing honest contents is preceded, which is for rewarding peers with high probability of providing honest contents by positioning peers with high probabilities of providing honest contents and holding desired contents at neighbors. As a result of conducted simulations comparing existent methods, our method achieves reasonable performance regardless of the distribution of the probability of holding desired contents over all peers, the ratio of malicious peers, and the probability of providing polluted contents by malicious peers.

In our method, peers are evaluated with respect to only providing and holding contents. However, in practice, the behavior of query forwarding or taking neighbors is also significant for performance. Concretely, if a peer disregards its received query message or if a peer itself provides honest contents, but deliberately takes malicious peers as neighbors, query messages arrive at only unfavorble peers. Hence, extending our method to evaluate other peers with respect to query forwarding and taking neighbors by rewarding honest

peers is for future research.

Chapter 5

Conclusion

5.1 Summary of Contributions

This dissertation features on the P2P content sharing system that is the most known application of P2P systems and spreads widely. Dissemination of polluted contents is a major security issue for the P2P content sharing system. This dissertation proposes methods for prevention of polluted contents to use P2P content sharing system securely. Methods of trust value computation for the simple and the probabilistic peers are proposed to select trustworthy peer to obtain honest contents. In addition, a method of overlay construction is proposed for preventablity of polluted contents and search efficiency. Furthermore, it is confirmed by simulations of P2P content sharing system that each proposed method improves pereventability of polluted contents more than existent methods. This dissertation contributes secure use of the P2P content sharing system.

In Chapter 1, the wide spreadding P2P content sharing system is described. Furthermore, the necessity of prevention of polluted contents due to its openness, growth of popularity, and the large interference of legitimate content sharing by dissemination of polluted contents are described. In addition, approaches for prevention of polluted contents are described.

In Chapter 2, the simple peer behaviour is assumed. A method of trust value computation that achieves high performance regardless of the ratio of the number of dishonest raters. A simulation of content sharing is conducted by varying the ratio of dishonest raters and the simulation confirms that the proposed method performs better than existent methods when the ratio of dishonest raters is no more than 50%.

In Chapter 3, the probabilistic peer behavior is assumed. A method of trust value computation that achieves high performance regardless of the ra-

tio or the probability of rating dishonestly of peers. As a result of conducted simulations, the proposed method performs no less or better than existent methods regardless of the probability of rating dishonestly or providing polluted contents. The proposed method performs better when the ratio of dishonest raters is no more than 50%.

In Chapter 4, a method of constructing an overlay is proposed to achieve both preventability of polluted contents and search efficienct. In this method, when a peer selects its neighbors, it evaluates peers by the both probabilities of holding desired contents and providing honest contents while the probability of providing honest contents is preceded. As a result of conducted simulations, the method proposed in this dissertation achieves reasonable performance regardless of distribution of the probability of holoding desired contents over all peers, the ratio of malicious peers, and the probability of providing polluted contents by malicious peers.

5.2 Future Research

5.2.1 Handling Various Models of Peer Behavior

In this dissertation, only limited models of peers are examined. Behevior models for providing content, and rating in a reputation system are assumed to be simple or probabilistic. In previous works, two main types of further models are assumed. The first model has to do with changing behavior through time [36]. In our method, the assumed model is the probabilistic behavior model at its most complex. However, in practice, there can be malicious peers that gain a good reputation by honest behavior until the malicious peers suddenly begins to behave maliciously. By introducing the weights of time factor to computation of trust values (e.g., the newer the rating values, the more the weights), our proposed method of trust value computation is expected to handle not only probabilistic behavior, but also time-controlled behavior.

Behavior on forwarding query messages on an overlay is assumed to be honesty only for the purpose of this dissertation. In AGP [21], peers that neither forward nor respond received query messages are assumed and they appear to be practical. Because such peers stopo query messages from traveling, the line-up of responding peers is massively affected. Our method of topology adaptation in Chapter 4 concentrates on evaluating peers with respect to providing content to establish a proper method for prevention of polluted contents. By including behavior on forwarding query messages in evaluation items of peers, the impact of such peers is expected to degrade.

5.2.2 Efficient and Secure Computation of Weights

In Chapters 2 and 3, weights of rating values were computed using a chain of peers with common ratee that connecting own and other peers. Using an overlay where there is a link if peers at both ends which have common ratees, the rating values are expected to be shared efficiently. In our methods in Chapters 2 and 3, necessary rating values for trust value computation are only for weights are not 0, i.e., rating values by peers within a certain distance on overlay. Therefore, rating values appear to be collected efficiently by querying on the overlay. Designing a collecting method is of further interest. In addition, the computation of weights and the suppression dishonesty on the computation, e.g., boosting the value of weights dishonestly should also be investigated in the future.

Acknowledgments

I have been fortunate to have received support and assistance from many individuals during the course of this work. I would especially like to thank Professor Toru Fujiwara for his individual guidance, suggestions to this work, continuous support and encouragement.

I would also like to express my heartfelt appreciation to Professors Shojiro Nishio, Fumio Kishino, Koh Hosoda, Norihisa Komoda, and Shinji Shimojo for their individual advice and encouragement. I would like to express my gratitude to Associate Professor Yuuichi Teranishi and Yoshifumi Manabe, Ph.D., for their kind willingness to read and evaluate my dissertation.

I would also like to express my gratitude to Associate Professor Yasunori Ishihara and Assistant Professors Maki Yoshida and Shingo Okamura for insightful comments, suggestions, and warm encouragement.

I would like to thank Associate Professors Kenichi Baba and Susumu Date, Research Associate Professor Eisaku Sakane, Research Lecturer Kazunori Nozaki, Assistance Professor Kohei Ichikawa, Yoshimasa Ishi, and Tomoya Kawakami for constructive comments and warm encouragement at my former laboratory.

I appreciate the feedback offered by Associate Professors Kaname Harumoto and Seiichi Kato, Lecturer Toyokazu Akiyama, Susumu Takeuti, Ph.D., Mikio Yoshida, Hidenori Kanjo, and Assistant Professor Kenji Hashimoto.

I received generous support from people in Fujiwara and Shimojo Laboratories.

Finally, but by no means least, I am deeply grateful to my family for their extraordinarily tolerant and support.

Bibliography

[1] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 202–215, 2001.

[2] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao., "OceanStore: An architecture for global-scale persistent storage," in *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*, 2000.

[3] J. Zhang, L. Liu, L. Ramaswamy, and C. Pu, "PeerCast: Churn-resilient end system multicast on heterogeneous overlay networks," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 821–850, 2008.

[4] D. Hughes, G. Coulson, and J. Walkerdine, "Free riding on Gnutella revisited: The bell tolls?," *IEEE Distributed Systems Online*, vol. 6, no. 6, 2005. `http://doi.ieeecomputersociety.org/10.1109/MDSO.2005.31`.

[5] A. Singh, M. Castro, P. Druschel, and A. Rowstron, "Defending against eclipse attacks on overlay networks," in *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, no. 21, 2004.

[6] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Peer-to-Peer Systems, LNCS 2429*, pp. 261–269, 2002.

[7] E. Chien, "Malicious threats of peer-to-peer networking," 2003. `http://www.symantec.com/avcenter/reference/malicious.threats.pdf`.

[8] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in P2P file sharing system," in *Proceedings of the 24th Annual Joint Conference of*

the IEEE Computer and Communications Societies (INFOCOM 2005), vol. 2, pp. 1174–1185, 2005.

[9] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, and W. Zwaenepoe, "Denial-of-service resilience in peer-to-peer file sharing systems," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 38–49, 2005.

[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web (WWW 2003)*, pp. 640–651, 2003.

[11] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[12] H. Tian, S. Zou, W. Wang, and S. Cheng, "Constructing efficient peer-to-peer overlay topologies by adaptive connection establishment," *Computer Communications*, vol. 29, no. 17, pp. 3567–3579, 2006.

[13] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)*, pp. 161–172, 2001.

[15] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, 2004.

[16] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001, LNCS 2218*, pp. 329–350, 2001.

[17] Y. Zhu, X. Yang, and Y. Hu, "Making search efficient on gnutella-like p2p systems," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, 2005.

[18] E. Löser, S. Staab, and C. Tempich, "Semantic social overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 5–14, 2007.

[19] I. Martinovic, C. Leng, F. A. Zdarsky, A. Mauthe, R. Steinmetz, and J. B. Schmitt, "Self-protection in P2P networks: Choosing the right neighbourhood," in *Proceedings of the 1st International Workshop on Self-Organizing Systems (IWSOS 2006) and the 3rd International Workshop on New Trends in Network Architectures and Services (EuroNGI 2006), LNCS 4124*, pp. 23–33, 2006.

[20] T. Condie, S. D. Kamvar, and H. Garcia-Molina, "Adaptive peer-to-peer topologies," in *Proceedings of the 4th International Conference on Peer-to-Peer Computing (P2P '04)*, pp. 53–62, 2004.

[21] I. Pogkas, V. Kriakov, Z. Chen, and A. Delis, "Adaptive neighborhood selection in peer-to-peer networks based on content similarity and reputation," *Peer-to-Peer Networking and Applications*, vol. 2, no. 1, pp. 37–59, 2009.

[22] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003)*, vol. 3, pp. 2166–2176, 2003.

[23] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "Managing and sharing servants' reputations in P2P systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 840–854, 2003.

[24] Y. Ito and H. Kawano, "Evaluation of P2P contents distribution system with cryptographic trust chains," *DBSJ Letters*, vol. 6, no. 1, pp. 21–24, 2007.

[25] G. Swamynathan, B. Y. Zhao, K. C. Almeroth, and H. Zheng, "Globally decoupled reputations for large distributed networks," *Advances in Multimedia*, vol. 2007, Article ID 92485, 2007.

[26] A. A. Selcuk, E. Uzun, and M. R. Pariente, "A reputation-based trust management system for P2P networks," in *Proceedings of IEEE International Symposium on Cluster Computing and the Grid, 2004 (CCGrid 2004)*, pp. 251–258, 2004.

[27] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, 2007.

[28] J. R. Douceur, "The sybil attack," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, pp. 251–260, 2002.

[29] T. Sakai, K. Terada, and T. Araragi, "Robust online reputation mechanism by stochastic approximation," *IEICE Transactions on Information and Systems*, vol. J88-D1, no. 5, pp. 958–968, 2005.

[30] R. Jurca and B. Faltings, "Using CHI-scores to reward honest feedback from repeated interactions," in *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems (AAMAS '06)*, pp. 1233–1240, 2006.

[31] A. Jøsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, pp. 324–337, 2002.

[32] Z. Liang and W. Shi, "PET: A PErsonalized trust model with reputation and risk evaluation for P2P resource sharing," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005 (HICSS '05)*, 2005.

[33] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," in *Proceedings of the 7th International Workshop on Trust in Agent Societies at the 3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS '04)*, pp. 106–117, 2004.

[34] J. van der Merwe, D. Dawoud, and S. McDonald, "Fully self-organized peer-to-peer key management for mobile ad hoc networks," in *Proceedings of the 4th ACM workshop on Wireless security (WiSe '05)*, pp. 21–30, 2005.

[35] Z. Liang and W. Shi, "Analysis of ratings on trust inference in open environments," *Performance Evaluation*, vol. 65, no. 2, pp. 99–128, 2008.

[36] M. Srivatsa, L. Xiong, and L. Liu, "TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks," in *Proceedings of the 14th international conference on World Wide Web (WWW 2005)*, pp. 422–431, 2005.

[37] D. Tsoumakos and N. Roussopoulos, "A comparison of peer-to-peer search methods," in *Proceedings of the WebDB*, pp. 61–66, 2003.

[38] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th international conference on Supercomputing (ICS '02)*, pp. 84–95, ACM, 2002.

[39] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal multicast," *ACM Transactions on Computer Systems*, vol. 17, no. 2, pp. 41–88, 1999.

[40] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proceedings of the 11th international conference on Information and knowledge management (CIKM '02)*, pp. 300–307, 2002.

[41] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulié, "Peer-to-peer membership management for gossip-based protocols," *IEEE Transactions on Computers*, vol. 52, no. 2, pp. 139–149, 2003.

[42] N. Ganguly, G. Canright, and A. Deutsch, "Design of an efficient search algorithm for P2P networks using concepts from natural immune systems," in *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, pp. 492–500, 2004.

[43] A. Crespo and H. Garcia-Molina, "Semantic overlay networks for P2P systems," in *Agents and Peer-to-Peer Computing, LNCS3601*, pp. 1–13, 2005.

[44] M. T. Schlosser, T. Condie, and S. D. Kamvar, "Simulating a file-sharing P2P network," Tech. Rep. 2003-28, Stanford InfoLab, 2003.

[45] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, 2002.

[46] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the gnutella network," *IEEE Internet Computing*, vol. 6, no. 1, pp. 50–57, 2002.

[47] Y. Zhao, X. Hou, M. Yang, and Y. Dai, "Measurement study and application of social network in the Maze P2P file-sharing system," in *Proceedings of the 1st international conference on Scalable information systems (InfoScale '06)*, 2006.