Trustworthiness of Acquaintances Based on Access Control in Peer-to-Peer Overlay Networks

Yoshio Nakajima*, Kenichi Watanabe*, Valbona Leonald*,
Naohiro Hayashibara*, Tomoya Enokido† and Makoto Takizawa*
*Tokyo Denki University, Japan †Rissho University, Japan
E-mail *{nak, nabe, valbona, haya, taki}@takilab.k.dendai.ac.jp, †eno@ris.ac.jp

Abstract

Objects are distributed to peers in a P2P overlay network. Service supported by an object is modeled to be a set of methods and QoS. An acquaintance peer of a peer p is a peer about whose service the peer p knows and with which p can directly communicate. We define how *satisfiable* a requesting peer is for access requests by taking into account the authorization. Acquaintance peers of a peer p may hold inconsistent information on target peers since it takes time to propagate change information of the target peers and peers may be faulty. Hence, it is critical to discuss how much a peer can trust each acquaintance. We define the *trustworthiness* of each acquaintance by aggregating the satisfiability which is obtained through each interaction with the acquaintance. Each time a peer gets a new acquaintance and acquaintance information is changed, each peer keeps it in record. Due to the limited size of memory, the peer throws away information of less trustworthy acquaintances to make a space to store new acquaintance information. The trustworthiness of acquaintance of each peer is propagated in a peer-by-peer way while some acquaintance information is recorded in a peer. We evaluate how the trustworthiness of acquaintance is changing through interactions among peers.

1. Introduction

Various types and huge number of peer computers are interconnected and the membership is dynamically changed in a *peer-to-peer* (P2P) overlay network. An object is a unit of resource. A group of peers (*processes*) on peer computers are cooperating by manipulating objects and exchanging messages. Service supported by each object is characterized by types of methods and quality of service (QoS). An object is distributed to peers with various ways like downloading and caching [13, 14] in P2P networks.

A peer is classified according to types of service, *holder* peer where objects are stored, *manipulation* peer which are allowed to manipulate objects, and *authorization* peer which can grant access rights to other peers [12–15]. An *acquaintance* peer of a peer p_i is a peer p_j whose service p_i knows and with which p_i can directly communicate. A peer first asks its acquaintances to detect target peers which can manipulate a target object so as to satisfy an access request which the peer issues. Even if some peer holds a target object, the peer cannot to manipulate the object if the peer is not granted an *access right* (*permission*). If acquaintances which satisfy the access request are not detected, each acquaintance peer furthermore asks its acquaintances. Acquaintance concepts are so far discussed only to detect target peers holding target objects [2, 4]. The authors discuss how peers cooperate with each other to obtain required service, e.g. find a manipulation peer of a target object and then ask the peer to manipulate the object in the paper [12, 15].

If service supported by a peer is changed, the change information is propagated through acquaintances. However, it takes time to propagate the change of the service to every peer due to the scalability and openness of the P2P overlay network. Hence, some acquaintances of a peer may show obsolete and inconsistent information on target peers of a target object. In addition, acquaintances may not only stop by fault but also be arbitrarily faulty [6]. Hence, it is critical to discuss how much a peer trusts its acquaintance. A requesting peer p is satisfiable for each access request to *find* a target peer if a target peer is detected. However, if p is not granted an access right, peer p is not satisfiable to *manipulate* a target object, even if the peer p finds where the object exists. We define the *satisfiability* of each type of access request to find an object, manipulate an object, and grant an access right of an object. Thus, we define the *trustworthiness* of an acquaintance by aggregating the satisfiability of each access request obtained through each interaction. The acquaintance relations are propagated through peer-to-peer interactions. Each peer can admit only limited amount of the acquaintance relations. Obsolete and untrustworthy acquaintance relations are thrown away to make space to store new acquaintance relations. We implement the algorithm for detecting target peers updating and propagating trustworthiness on peers. We evaluate the peers in terms of hit ratio and number of messages.

In section 2, we present acquaintance relations of peers. In section 3, we discuss the trustworthiness of an acquaintance. In section 4, we discuss how to implement peers. In section 5, we evaluate the peers.

2 Acquaintances

In P2P overlay networks [1, 7–10], it is discussed only how to detect a target peer with a target object. Even if a *target* object is detected, the object cannot be manipulated if the requesting peer is not authorized. An *access right* (on *permission*) is specified in a form $[o, op]$ for an object o and a method op [3]. An access request to manipulate an object o through a method op is written in a from $\langle o, op \rangle$.

Only if a peer p is granted an access right $[o, op]$, an access request $\langle o, op\rangle$ issued by p can be accepted.

First, an application issues an access request $\langle o, op\rangle$ to a local peer p. On receipt of request $\langle o, op\rangle$, a peer has to find target peers of the access request. It may be impossible for each peer to perceive what service of what objects each peer supports due to the scalability. In addition, the type and quality of service supported by each peer, and the membership of a P2P overlay network are dynamically changed. Each peer is in an *acquaintance* relation with another peer, and the peers exchange service information of their acquaintances with each other. Information on the type and quality of service of each peer is propagated in the network. A peer makes a decision on which acquaintances the peer issues an access request based on the information obtained from the acquaintances. If a peer p_i issues an access request to another peer p_j, p_i and p_j are requesting and *requested* peers, respectively. There are the following types of peer-to-object (P2O) relations [13, 14]:

a. A peer p *holds* an object o (written as $p \mid o$).
b. A peer p *can manipulate* an object o through a method op ($p \models_{op} o$), i.e. p is granted an access right $[o, op]$. Here, p is a *manipulation* peer of an access request $\langle o, op\rangle$. p is a *surrogate* peer of $\langle o, op\rangle$ ($p \mapsto_{op} o$) if $p \models_{op} o$ and p can issue $\langle o, op\rangle$ on behalf of a requesting peer.
c. A peer p *can grant* an access right $[o, op]$ to another peer ($p \vdash_{op} o$). Here, p is an *authorization* peer.
d. A peer p is a *direct manipulation* peer of an access request $\langle o, op\rangle$ ($p \triangleright_{op} o$) iff $p \mid o$ and $p \models_{op} o$.
e. A peer p is a *servicing* peer of $\langle o, op\rangle$ ($p \square_{op} o$) iff $p \mid o$, $p \models_{op} o$, $p \mapsto_{op} o$, or $p \vdash_{op} o$.
f. For a peer p and an object o, $p \models o$, $p \vdash o$, $p \square o$, and $p \mapsto o$ iff $p \models_{op} o$, $p \vdash_{op} o$, $p \square_{op} o$, and $p \mapsto_{op} o$, respectively, for some method op.

We define the acquaintance relations $\rightarrow$ on the P2O relation $\square$ for a peer p, object o, and method op as follows:

- $p \rightarrow (p_i \square_{op} o)$ iff a peer p perceives $p_i \square_{op} o$.
- $p \rightarrow^* (p_i \square_{op} o)$ iff $p \rightarrow (p_i \square_{op} o)$ or $p \rightarrow (p_k \rightarrow^* (p_i \square_{op} o)$ for some peer p_k.
- $p \rightarrow (p_i \square o)$ and $p \rightarrow^* (p_i \square o)$ iff $p \rightarrow (p_i \square_{op} o)$ and $p \rightarrow^* (p_i \square_{op} o)$ for some method op, respectively.

If $p_i \rightarrow^* (p_j \mid o)$, $p_i \rightarrow^* (p_j \models o)$, $p_i \rightarrow^* (p_j \mapsto o)$, and $p_i \rightarrow^* (p_j \vdash o)$, a peer p_j is a *holder*, *manipulation*, *surrogate*, and *authorization* acquaintance of an object o, respectively. Let $view(p_i)$ be a set of acquaintance peers of a peer p_i.

3. Trustworthiness of Acquaintance Peer

3.1 Satisfiability of access request

A peer may lose objects and obtain new acquaintances. Thus, the P2O relations are changed in a P2P overlay network. An acquaintance peer p_j of a peer p_i may not hold the same information of a target object as one which p_i has previously obtained from p_j because it takes time to propagate change information in networks. Thus, some acquaintance peer maintains up-to-date information of a target peer but another acquaintance holds obsolete inconsistent one. Hence, each peer p_i has to recognize what acquaintance peers can be trusted by p_i.

Suppose a peer p_i issues a manipulation request $\langle o, \models, op\rangle$ to another peer p_j for manipulating an object o through a method op as discussed in the preceding section. First, suppose the peer p_j is granted an access right $[o,op]$ ($p_j \models_{op} o$). The peer p_j locally manipulates the object o if p_j is a holder of an object o ($p_j \mid o$), i.e. $p_j \triangleright_{op} o$. Then, p_j sends the reply r_i to the requesting peer p_i. Here, p_i is satisfied because p_i can obtain the result for the access request $\langle o, op\rangle$. Unless $p_j \mid o$, one of the peers p_i and p_j has to detect a holder peer. Here, suppose p_i asks the acquaintance p_j to detect a holder peer and p_j finds a holder peer p_k in the acquaintances. The manipulation peer p_j issues the manipulation request $\langle o, \models, op\rangle$ to the holder peer p_k since $p_j \models_{op} o$. Here, p_i is less satisfiable since p_i cannot directly get the result from the acquaintance p_j.

We define the *satisfiability* $\sigma_{ij}(\langle o, \square, op\rangle)$ of a peer p_i to an acquaintance peer p_j in terms of type of network access request $\langle o, \square, op\rangle$ and states of the peers p_i and p_j. $state(p_i)$ shows how the peer p_i is related with an object o with respect to a method op, i.e. $p_i \mid o$, $p_i \models_{op} o$, and $p_i \vdash_{op} o$. Table 1 summarizes the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$. Suppose that a peer p_i whose state is $p_i \models_{op} o$ and $p_i \not\mid o$ issues an access request $\langle o, |, \_\rangle$ to another peer p_j. If $p_i | o$, p_i is the most satisfiable. Here, $\sigma_{ij}(\langle o, |, \_\rangle) = 1$. Next, if $p_j \not\mid o$ but p_j knows another peer p_k is a holder, i.e. $p_j \rightarrow (p_k \mid o)$, p_i cannot get the result from p_j but may get the result from p_k. Here, $\sigma_{ij}(\langle o, |, \_\rangle) = \delta_i$, where δ_i is a *distance factor* showing how friendly and open-minded a peer p_i is for another peer. $0 \leq \delta_i \leq 1$. "$\delta_i = 1$" means the a peer p_i is open-minded, i.e. p_i can always ask another peer p_j if p_i knows p_j. "$\delta_i = 0$" shows that p_i dislikes to ask another peer.

3.2 Trustworthiness

The *trustworthiness* $\tau_{ij}(\langle o, \square, op\rangle)$ of a peer p_i to an acquaintance peer p_j for an access request $\langle o, \square, op\rangle$ is obtained by aggregating the satisfiability of each access request issued to p_j. The peer p_i keeps in record. The satisfiability σ_{ij} obtained at each interaction with each acquaintance p_j. The trustworthiness is calculated by the function: $Trust0(\tau, \sigma, \alpha) = \alpha \cdot \tau + (1 - \alpha) \cdot \sigma$. Suppose a peer p_i obtains the satisfiability σ_{ij} for an access request $\langle o, \square, op\rangle$ from an acquaintance peer p_j. Let τ_{ij} be the current trustworthiness of the peer p_i to p_j on $\langle o,\square,op\rangle$. The trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ is changed with $Trust0(\tau_{ij}, \sigma_{ij}, \alpha_i)$. Initially, $\tau_{ij}(\langle o, \square, op\rangle)$ is defined as 0. Here, α_i is a constant ($0 \leq \alpha_i \leq 1$) for a peer p_i. If $\alpha_i = 1$, the trustworthiness is not changed even if new satisfiability is obtained. If $\alpha_i = 0$, the trustworthiness is decided only by the satisfiability. The smaller α_i is, the more important the satisfiability obtained for a current request $\langle o, \square, op\rangle$ is.

Suppose a peer p_i issues an access request $\langle o, \square, op\rangle$ to

Table 1. Satisfiability $\sigma_{ij}(\langle o, \Box, op\rangle)$.

state of p_i	network access requests	states of p_i's acquaintances	satisfiability
$p_i \mid o$ and $p_i \models_{op} o$	$\langle o, op\rangle$	-	$\sigma_{ii} = 1$
$p_i \models_{op} o$ and $p_i \not\mid o$	$\langle o, \mid, \_\rangle$	$p_j \mid o$	$\sigma_{ij} = 1$
		$p_j \rightarrow (p_k \mid o)$	$\sigma_{ij} = \delta_i, \sigma_{ik} = 1$
$p_i \mid o$ and $p_i \not\models_{op} o$	$\langle o, \vdash, op\rangle$	$p_j \vdash_{op} o$	$\sigma_{ij} = 1$
	$\langle o, \models, op\rangle$	$p_j \mapsto_{op} o$	$\sigma_{ij} = 1$
$p_i \vdash_{op} o$ and $p_i \not\mid o$	$\langle o, \mid, \_\rangle$	$p_j \mid o$	$\sigma_{ij} = 1$
		$p_j \rightarrow (p_k \mid o)$	$\sigma_{ij} = \delta_i, \sigma_{ik} = 1$
$p_i \not\vdash_{op} o$	$\langle o, \models, op\rangle$	$p_j \models_{op} o$	$\sigma_{ij} = 1$
	$\langle o, \vdash, op\rangle$, $\langle o, \mid, \_\rangle$	$p_j \vdash_{op} o$, $p_k \mid o$	$\sigma_{ij} = \delta_i, \sigma_{ik} = \delta_i$
	$\langle o, \models, op\rangle$, $\langle o, \mid, \_\rangle$	$p_j \models_{op} o$, $p_k \mid o$	$\sigma_{ij} = \delta_i, \sigma_{ik} = \delta_i$

another peer p_j. Here, the peer p_j does not support the P2O relation $p_j \Box_{op} o$ but p_j perceives that some peer p_k supports the required service, i.e. $p_j \not\Box_{op} o$ and $p_j \rightarrow (p_k \Box_{op} o)$. On receipt of the request from p_i, the peer p_j informs p_i of "p_k $\Box_{op}$"" o. Here p_j is referred to as *informing* peer of p_k. There are two choices, the requesting peer p_i directly manipulates p_k or p_i asks the acquaintance p_j to manipulate p_k. Suppose p_i directly issues an access request $\langle o, \Box, op\rangle$ to p_k. If p_i receives the reply from p_k, the satisfiability $\sigma_{ik}(\langle o, \Box, op\rangle)$ is obtained from Table 1. Here, the trustworthiness is calculated by the function: $Trust1(\tau, \sigma, \beta) = [\beta + (1\text{-}\beta)\cdot\sigma]\tau$. The trustworthiness $\tau_{ik}(\langle o, \Box, op\rangle)$ is changed with $Trust0(\tau_{ik}(\langle o, \Box, op\rangle), \sigma_{ik}(\langle o, \Box, op\rangle), \alpha_i)$ as discussed here. In addition, $\tau_{ij}(\langle o, \Box, op\rangle)$ to the informing peer p_j is also changed. Let τ_{ij} be the current trustworthiness of a peer p_i to a peer p_k and σ_{ik} be the satisfiability of an access request $\langle o, \Box, op\rangle$ issued to the peer p_k. The trustworthiness $\tau_{ij}(\langle o, \Box, op\rangle)$ is changed with $Trust1(\tau_{ij}, \sigma_{ik}, \beta_i)$. Here, β_i is a constant defined for a peer p_i and $0 \leq \beta_i \leq 1$. The smaller β_i is, the more the satisfiability $\sigma_{ik}(\langle o, \Box, op\rangle)$ dominates the trustworthiness $\tau_{ij}(\langle o, \Box, op\rangle)$. This means, $\tau_{ij}(\langle o, \Box, op\rangle)$ is decreased if the peer p_j introduces a less trustworthy peer p_k to p_i.

3.3 Ranking factors

The reputation [5] of a peer p_j shows how much an acquaintance peer p_j of a peer p_i is trusted by other peers. In this paper, each peer p_i only takes into account how much its trustworthy acquaintance peer trusts the acquaintance peer p_j. We introduce the *ranking* factor $\rho_{ij}(\langle o, \Box, op\rangle)$ to show how much an acquaintance peer p_j of a peer p_i is trusted for an access request $\langle o, \Box, op\rangle$. In this paper, $\rho_{ij}(\langle o, \Box, op\rangle)$ depends on how much a trustworthy acquaintance peer p_k of p_i trusts p_j, i.e. $\tau_{ik}(\langle o, \Box, op\rangle) \cdot \tau_{kj}(\langle o, \Box, op\rangle)$ [12]. Suppose there are six peers p_0, p_1, p_2, p_3, p_4, and p_5 where $view(p_0) = \{p_1, p_2, p_3, p_4\}$ and $view(p_1) = \{p_2, p_3, p_4, p_5\}$. Suppose the trustworthiness for each peer is given as $\tau_{02}(\langle o, \models, op\rangle) = 0.7$, $\tau_{03}(\langle o, \models, op\rangle) = 0.3$, $\tau_{04}(\langle o, \models, op\rangle) = 0.4$, $\tau_{21}(\langle o, \models, op\rangle) = 0.8$, $\tau_{31}(\langle o, \models, op\rangle) = 0.5$, $\tau_{41}(\langle o, \models, op\rangle) = 0.6$, and $\tau_{51}(\langle o, \models, op\rangle) = 0.5$. Here, the ranking factor $\rho_{01}(\langle o, \models, op\rangle)$ to p_1 is $(0.8\cdot 0.7 + 0.5\cdot 0.3 + 0.6\cdot 0.4)/3 = 0.317$. The trustworthiness $\tau_{51}(\langle o, \models, op\rangle)$ is not considered in the ranking factor ρ_{01} since p_5 is not an acquaintance peer of p_0. According to the traditional reputation concepts [16], the ranking factor ρ_{01} is given as $(\tau_{21} + \tau_{31} + \tau_{41} + \tau_{51})/4 = (0.8 + 0.5 + 0.6 + 0.5)/4 = 0.6$. If p_5 is not trustworthy for p_0, e.g. p_5 is malicious, ρ_{01} is not reliable. Only the trustworthiness of a trustworthy acquaintance peer is considered. In the paper [15], we show how to shake off the trustworthiness from an untrustworthy peer.

Let τ_{ij} and σ_{ij} stand for $\tau_{ij}(\langle o,\Box,op\rangle)$ and $\sigma_{ij}(\langle o,\Box,op\rangle)$ for an access request $\langle o,\Box,op\rangle$, respectively, for simplicity. Each peer p_k distributes the trustworthiness τ_{kj} for every acquaintance peer p_j to every acquaintance peer in the view $view(p_k)$. Each peer p_i calculates the ranking factor ρ_{ij} by using the formula $Rank(p_i, p_j, \langle o, \Box, op\rangle) = \sum_{p_k \in view(p_i)} \tau_{ik} \cdot \tau_{kj} \,/\, |\{ p_k \in view(p_i) \mid \tau_{ik} \cdot \tau_{kj} \neq 0\}|$. The ranking factor $\rho_{ij}(\langle o, \Box, op\rangle)$ is changed with $Rank(p_i, p_j, \langle o, \Box, op\rangle)$ after updating the trustworthiness information in AB_i.

4 Implementation

4.1 Acquaintance bases

Each peer p_i maintains an acquaintance base AB_i to store the view $view(p_i)$ and acquaintance information obtained from the acquaintances. A scheme of AB_i is given a tuple $\langle pid, sid, oid, op, req, \sigma, \tau, \rho, \{iid\}\rangle$ of attributes. Here, an attribute *pid* shows an identifier of an acquaintance of p_i. *oid* is an identifier of an object, *req* is a type $\Box$ of access request $\in \{\mid, \vdash, \models\}$. *sid* is an identifier of a peer which supports service satisfying the request *req* on the object *oid*. *op* is a method. σ, τ, and ρ are the satisfiability, trustworthiness, and ranking factor of p_i to the acquaintance peer *pid*, respectively. *iid* shows a set of informing peers which informs p_i of the acquaintance information. Suppose a peer p_i newly obtains an acquaintance peer p_j which is a target peer of an access request $\langle o,\Box,op\rangle$, i.e. $p_i \rightarrow (p_j \Box_{op} o)$. A tuple $\langle p_j, p_j, o, op, \Box, \sigma_{ij}, \tau_{ij}, \rho_{ij}, \_\rangle$ is stored in AB_i.

Here, $\sigma_{ij} = \delta_i$ and $\tau_{ij} = Trust0(0, \sigma_{ij}, \alpha_i) = (1 - \alpha_i)\sigma_{ij}$. The ranking factor ρ_{ij} is obtained by $Rank(p_i, p_j, \langle o,\Box,op\rangle)$.

Next, suppose a peer p_j is an acquaintance of a peer p_k where $p_k \Box_{op} o$ and sends acquaintance information $p_j \rightarrow (p_k \Box_{op} o)$ to a peer p_i. On receipt of the acquaintance information from p_k, a tuple $\langle p_j, p_k, o, op, \Box, \sigma_{ij}, \tau_{ij}, \rho_{ij}, \_\rangle$ is stored in AB_i, where $\sigma_{ij} = \delta_i \cdot \sigma_{jk}$, $\tau_{ij} = Trust0(\tau_{ij}, \sigma_{ij}, \alpha_i) = \alpha_i \cdot \tau_{ij} + (1 - \alpha)\cdot\sigma_{ij}$, and $\rho_{ij} = Rank(p_i, p_j, \langle o,\Box,op\rangle)$. In addition, a tuple $\langle p_k, p_k, o, op, \Box, \sigma_{ik}, \tau_{ik}, \rho_{ik}, \{p_j\}\rangle$ is stored in AB_i. If the trustworthiness in the tuple is updated, the trustworthiness of the tuple of p_k is also updated. In the manipulation, the informing peer p_j in the tuple is removed after it takes time. Here, $\sigma_{ik} = \delta_i \cdot \sigma_{jk}$, $\tau_{ik} = (1 - \alpha_i)\cdot\sigma_{ik}$, and $\rho_{ik} = Rank(p_i, p_j, \langle o,\Box,op\rangle)$. Suppose p_i issues an access request $\langle o, \Box, op\rangle$ to p_k by using the acquaintance information tuple $\langle p_k, p_k, o_h, op, \Box, \sigma_{ik}, \tau_{ik}, \rho_{ik}, \{p_j\}\rangle$ in AB_i. Then, p_i receives the reply from p_k and obtains the satisfiability σ. Here, the tuple is updated as $\sigma_{ik} = \sigma$ and τ_{ik} is changed with $Trust0(\tau_{ik}, \sigma, \alpha_i) = (1 - \alpha_i)\tau_{ik} + \alpha_i \cdot \sigma$. The ranking factor ρ_{ik} is changed with $Rank(p_i, p_j, \langle o,\Box,op\rangle)$. In addition, τ_{ij} of a tuple $\langle p_j, p_k, o, op, \Box, \sigma_{ij}, \tau_{ij}, \rho_{ij}, \{p_l\}\rangle$ in AB_i is changed with $Trust1(\tau_{ij}, \sigma_{ik}, \beta_i) = [\beta_i + (1 - \beta_i)\sigma_{ik}] \cdot \tau_{ij}$. If $iid \neq \phi$, τ_{il} of $\langle p_l, \cdots, \tau_{il}, \cdots\rangle$ in AB_i is also changed for every peer p_l in iid since p_j is introduced to p_i by p_l as discussed here.

4.2 Inter-peer communication

A peer communicates with acquaintances by exchanging request and reply messages. Suppose a peer p_i sends an access request $\langle o, \Box, op\rangle$ to an acquaintance peer p_j. A request message q is composed of the following fields:

- $q.id$ = unique identifier of the message q.
- $q.src$ = requesting peer p_i.
- $q.TTL$ = TTL (time to live) of the message q.
- $q.oid$ = identifier of the target object o.
- $q.op$ = method op.
- $q.atype$ $\triangle$ = type of access request.

In this paper, we assume there is some mechanism to assign a unique identifier to each message. Each time a message m passes a peer, $m.TTL$ is decremented by one. If $m.TTL = 0$, m is discarded.

Suppose that a peer p_i receives a request message q for an access request $\langle o, \Box, op\rangle$ from an acquaintance p_j. The peer p_i checks if p_i supports service required by the access request $\langle o_h, \Box, op\rangle$. If supported, i.e. $p_i \Box_{op} o$, p_i sends a *reply* message r to the requesting peer p_j. Otherwise, p_i forwards the request ϕ to the acquaintances. A *reply* message r includes the following fields:

- $r.id$ = identifier of the reply message r.
- $r.src$ = source peer which sends r.
- $r.qid$ = identifier $q.id$ of the access request q, i.e. r is a reply of the request q.
- $r.oid$ = identifier of the target object, $r.oid = q.oid$.
- $r.sid$ = identifier of the target peer.
- $r.\sigma$ = satisfiability of p_i to the target peer $r.sid$.
- $r.\tau$ = trustworthiness of p_i to the peer $r.sid$.
- $r.\rho$ = ranking factor of p_i to the peer $r.sid$.

Since the peer p_i is a target peer of the object o, p_i sends the requesting peer p_j a reply message r such that $r.oid = o$, $r.sid = p_i$, and $r.\sigma = \sigma_{ii} = 1$. If $p_i \not\Box_{op} o$, p_i searches the acquaintance base AB_i for tuples of the access request $\langle o,\Box,op\rangle$. Suppose a tuple $\langle p_j, p_k, o, op, \Box, \sigma_{ij}, \tau_{ij}, \rho_{ij}, p_f\rangle$ is found in AB_i. Here, $j = k$ if $p_i \rightarrow (p_k \Box_{op} o)$. If $p_i \rightarrow (p_j \rightarrow (p_k | o))$, $j \neq k$. The peer p_i sends a *reply* message r to the requesting peer p_j where $r.sid = p_k$, $r.\sigma = \sigma_{ij}$, $r.\tau = \tau_{ij}$, and $r.\rho = \rho_{ij}$.

If not found in AB_i, p_i decrements $q.TTL$ by one. If $d.TTL \geq 1$, p_i forwards the access request q to every acquaintance peer. The peer p_i waits for a reply from the acquaintance peers. If $q.TTL = 0$, p_i discards q.

On receipt of a reply message r of the request q from p_j, a peer p_i updates AB_i as follows:

1. If a tuple $\langle p_j, p_k, o_h, op, \Box, \sigma_{ij}, \tau_{ij}, \rho_{ij}, p_f\rangle$ is found in AB_i, σ_{ij}, τ_{ij}, and ρ_{ij} are replaced with $r.\sigma$, $\alpha_i \cdot \tau_{ij} + (1 - \alpha_i)\cdot\sigma_{ij}$, and $Rank(p_i, p_j, \langle o,\Box,op\rangle)$, respectively.
2. If $p_f \neq$ "_", the trustworthiness τ_{if} of p_i to p_f is also updated as discussed here.
3. If $\langle p_j, p_k, o, op, \Box, \sigma_{ij}, \tau_{ij}, \rho_{ij}, p_l\rangle$ is not found, a tuple $\langle p_j, r.sid, r.oid, q.op, q.atype, r.\sigma, r.\tau, \rho, \_\rangle$ is added to AB_i where $\rho = Rank(p_i, p_j, \langle o,\Box,op\rangle)$.

The peer p_i waits for a reply message from every acquaintance peer which p_i sends a request message q. If p_i receives every reply message or the timer expires, p_i takes a reply message r whose satisfiability is the largest, out of the reply messages received. The peer p_i sends the reply message of the request q to the requesting peer p_j.

On receipt of a reply message r showing $p_j \rightarrow (p_k \Box_{op} o)$ from an acquaintance peer p_j, p_i perceives that a peer p_k is a target peer of the target object o for the acquaintance p_j. The peer p_i cannot just take the target peer p_k as an acquaintance peer of p_i, i.e. a tuple $\langle p_k, p_k, o, op, \Box, \sigma_{ik}, \tau_{ik}, \rho_{ik}, p_j\rangle$ where $\sigma_{ik} = \delta_i$, $\tau_{ik} = \sigma_{ik}$, and $\rho_{ik} = Rank(p_i, p_j, \langle o,\Box,op\rangle)$. Because p_k might not intend to directly communicate with p_i. That is, the target object o cannot be obtained by p_i without asking the acquaintance peer p_j. One way is that p_i sends an *invitation* message to p_k. If p_k accepts the invitation to be an acquaintance of p_i, p_k sends an *accepted* message to p_i. Here, p_i adds a tuple $\langle p_k, p_k, o, op, \Box, \sigma_{ik}, \tau_{ik}, \rho_{ik}, p_j\rangle$ to AB_i. This is a *polite* way. In another way, p_i unilaterally recognizes p_k as its acquaintance if p_i receives the information $p_k \Box_{op} o$ from another peer p_j. Here, $\langle p_k, p_k, o, op, \Box, \sigma_{jk}, \tau_{jk}, \rho_{jk}, p_j\rangle$ is added to AB_i. Then, p_i may send a request $\langle o,\Box,op\rangle$ to p_k. If p_k rejects the request from p_i, σ_{ik}, τ_{ik}, and ρ_{ik} are decreased and p_i asks p_j to be an acquaintance.

The acquaintance base AB_i can include only a limited number t_i of tuples. Suppose a peer p_i would like to add a tuple a into AB_i. If AB_i is full, the tuple a cannot be added to AB_i. Here, a tuple b in AB_i is selected and removed to make a space to store the tuple a by the following rule:

[Selection rule]

1. Select a tuple b where $b.\tau$ is the smallest in AB_i.
2. if there are multiple tuples at step 1, select a tuple b where $b.\rho$ is the smallest in the tuples.

3. **if** there are still multiple tuples at step 2, select a tuple b where $b.\sigma$ is the smallest in the tuples is selected.

[Maintenance of AB_i] On recept of a *reply* message r from an acquaintance p_j, p_i obtains acquaintance information:

if $p_i \rightarrow (p_j \Box_{op} o)$, {
  $\sigma_{ij} = r.\sigma \cdot \delta_{ij}$; **stAB**($p_i$, $\langle p_j, p_k, o, op, \Box, \sigma_{ij} \cdot \delta_i$, *Trust0*(0, σ_{ij}, α_i), 0, _, 0)); }
if $p_i \rightarrow (p_j \rightarrow (p_k \Box_{op} o))$, {
  $\sigma_{ij} = r.\sigma \cdot \delta_i$; **stAB**($p_i$, $\langle p_j, p_k, o, op, \Box, \sigma_{ij}$, *Trust0*(0, σ_{ij}, α_i), 0, p_j, 0));
  if p_i is not careful, {
    $\sigma_{ik} = \sigma_{ij}$;
    stAB(p_i, $\langle p_k, p_k, o, op, \Box, \sigma_{ik}$, *Trust0*(0, σ_{ik}, α_i), 0, p_j, 0));
} }

stAB(p_i, $\langle p_j, p_k, o, op, \Box, \sigma, \tau, \rho, p_f, c \rangle$) {
  if (t = **findAB**(p_j, o, op, $\Box$)) $\neq$ NULL), {
    $\sigma_{ij} = t.\sigma \cdot \delta_i$;
    upAB(p_i, t, σ_{ij}, *Trust0*($t.\tau$, σ_{ij}, α_i), $t.\rho$, o, op, $\Box$, $t.iid \cup \{p_f\}$);
  } **else** {
    if AB_i is **full**, {
      one tuple is selected and removed;
      $\langle p_j, p_k, o, op, \Box, o, \tau, \rho, p_f, 0 \rangle$ is stored is AB_i;
  }}
  if $iid = \phi$, **return**;
  for every p_k **in** iid, {
    u = **findAB**(p_k, o, op, $\Box$);
    if $u \neq$ NULL, {
      τ_{ik} = *Trust1*($u.\tau$, σ_{ij}, β_i);
      ρ_{ik} = *Round*(p_i, p_k, $\langle o, \Box, op \rangle$);
      upAB(p_k, t, $u.\sigma$, τ_{ik}, ρ_{ik}, o, op, $\Box$, $u.iid$);
} } }

upAB (p_j, t, σ_{ij}, τ_{ij}, ρ_{ij}, o, $\Box$, op, iid) {
  $t.\sigma = \sigma_{ij}$; $t.\tau = \tau_{ij}$; $t.\rho = \rho_{ij}$; $t.iid = iid$; }

findAB (p_i, o, op, $\Box$) {
  if $t = \langle p_i, p_j, o, op, \Box, \cdots, c \rangle$ is found in AB_i, {
    $t.c = t.c + 1$; **return** (t);
  } **else return** ($NULL$); }

5 Evaluation

Each peer is realized as a Java process in the distributed simulation Neko [11]. A P2P overlay network includes n ($\geq$1) peers $p_1, \cdots, p_n$. Initially, each peer p_i is in an acquaintance relation with l_i ($\leq n$) peers which are randomly selected. There are m ($\geq$1) objects $o_1, \cdots, o_m$. Each object o_h is randomly distributed to some number of peers. Here, the distribution ratio ζ_h is the ratio of the number l_h of peers each of which holds a replica of an object o_h to the total number n of the peers, $\zeta_h = l_h / n$. The acquaintance base AB_i of each peer p_i can admit at most t_i tuples.

In the simulation, one peer p_i is randomly selected as a requesting peer and an object o_h is also randomly selected as a target object. We consider a detection request in the evaluation and a simple flooding algorithm to send the detection request. The selected peer p_i sends a *detection* request $\langle o_h, |, - \rangle$ message to every acquaintance peer of p_i to find target holder peers of the object o_h. This is the first round. Then, one requesting peer and a target object are randomly selected again. The requesting peer issues the detecting request as presented in the first round. This is the second round. In each round, the acquaintance bases (AB_i) of peers are changed as discussed. Hence, acquaintance information is distributed to the more number of peers after more rounds. However, since the volume of AB_i of each peer p_i is limited, some acquaintance information might be lost due to the tuple replacement. Here, some acquaintance peer may hold inconsistent acquaintance information. A sequence of rounds is referred to as one *run*. In this evaluation, totally 100 runs are performed.

In the evaluation, we assume that there are 1000 peers, i.e. n = 1000. Each peer p_i is initially related with three acquaintances, i.e. l_i = 3. We assume each peer p_i can store at most five tuples in AB_i, i.e. $t_i = t$ = 5. We assume $\tau_i = \tau$ for every peer p_i. The distant factor δ_i for each peer p_i is assumed to be 0.5, $\alpha_i = \alpha$ = 0.9, and $\beta_i = \beta$ = 0.9 for every peer p_i. TTL is 7. We assume $\zeta_h = \zeta$ for every object o_h.

First, we measure the hit ratio and the satisfiability for one object, i.e. m = 1. The hit ratio for an access request is defined to be probability that a target peer is detected. For the kth round, the number s ($\leq$100) of runs where a target peer is detected are obtained in the 100 runs. Then, the hit ratio of the kth round is given as s / 100.

Figure 1 shows the hit ratio for ζ = 1 [%] and ζ = 10 [%]. The horizontal axis shows the number of rounds.

Through interactions among peers, acquaintance information is propagated in the network. At the more rounds, the higher the satisfiability must be. Figure 2 shows the satisfiability for ζ = 1 [%] and 10 [%].

Tuples in the acquaintance base AB_i of each peer p_i are replaced with new tuples. Figures 3 and 4 show the hit ratio and satisfiability for sizes of the acquaintance base for t = 3,5, and 10 tuples.

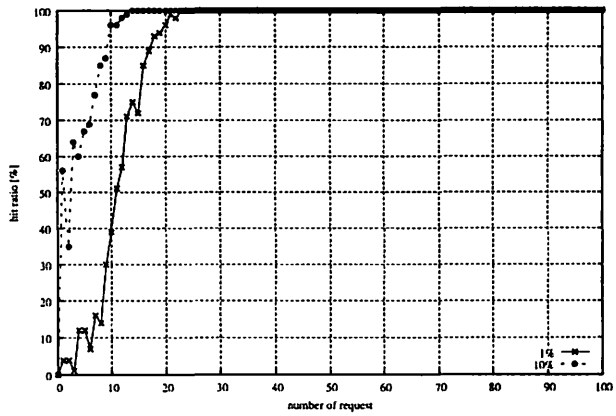

Figure 1. Hit ratio.

6 Concluding Remarks

We discussed how each peer trusts acquaintances in a peer-to-peer (P2P) overlay network. First, types of acquaintance relations are defined with respect to types of service of each peer. In addition to finding where a target object exists, a requesting peer has to find an authorization acquaintance to obtain the access right and a manipulation peer which

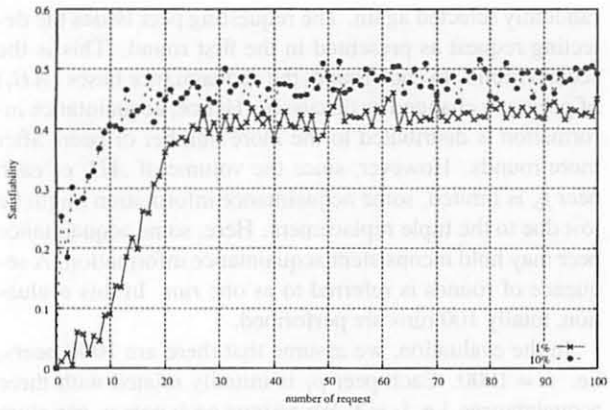

Figure 2. Satisfiability.

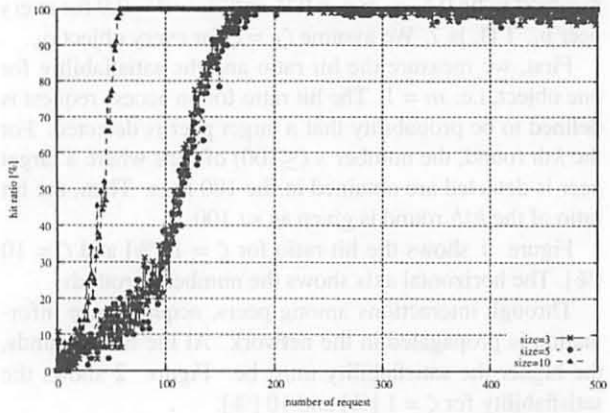

Figure 3. Hit ratio for acquaintance base size.

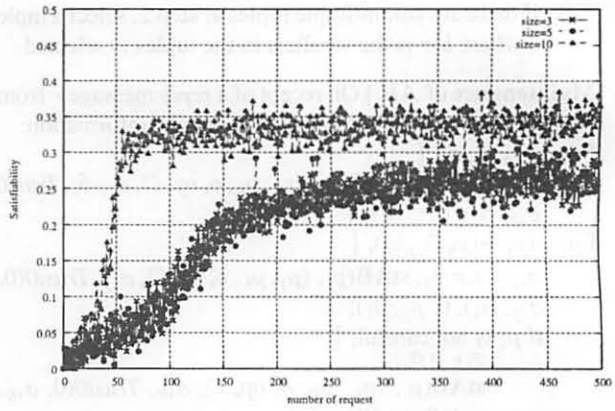

Figure 4. Satisfiability for acquaintance base size.

can manipulate the target object. Based on the acquaintance relations, we defined the satisfiability of an access request issued to an acquaintance peer in terms of types of service. Then, we defined the trustworthiness of each acquaintance and the ranking factor of each peer by aggregating the satisfiability obtained through each interaction with the acquaintance. We discussed how each peer behaves to obtain the trustworthiness and ranking factor by issuing access request to and receiving replaces from acquaintances. We evaluated how the trustworthiness and satisfiability of acquaintances are changing through interactions among peers.

References

[1] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Proc. of the Workshop on Design Issues in Anonymity and Unobservability*, pages 311–320, 2000.

[2] A. Crespo and H. Garcia-Molina. Routing Indices for Peer-to-Peer Systems. In *Proc. of the 22nd IEEE ICDCS*, pages 23–32, 2002.

[3] D. E. Denning and P. J. Denning. Data security. In *ACM Computing Surveys (CSUR)*, pages 227 – 249, 1979.

[4] T. Egemen, N. Deepa, and S. Hanan. An efficient nearest neighbor algorithm for P2P settings. In *Proceedings of the 2005 National Conference on Digital Government Research.*, pages 21–28, May 2005.

[5] D. S. Kamvar, T. M. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the 12th IEEE International Conference on World Wide Web*, pages 640–651, 2003.

[6] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. In *ACM Transactions on Programming Languages and Systems*, pages 382–401, 1982.

[7] Napster. http://www.napster.com.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proc. of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 161–172, 2001.

[9] M. Ripeanu. Peer-to-Peer Architecture Case Study: Gnutella Network. In *Proc. of International Conference on Peer-to-Peer Computing (P2P2001)*, pages 99–100, 2001.

[10] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems. In *Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001.

[11] P. Urban, X. Defago, and A. Schiper. Neko: A single environment to simulate and prototype distributed algorithms. *In Proc. of the 15th Int'l Conf. on Information Networking (ICOIN-15)*, pages 503–511, February 2001.

[12] K. Watanabe, T. Enokido, and M. Takizawa. Trustworthiness of acquaintances in peer-to-peer overlay networks. *accepted at International Journal of High Performance Computing and Networking (IJHPCN)*, 2006.

[13] K. Watanabe, T. Enokido, M. Takizawa, and K. Kim. Charge-based Flooding Algorithm for Detecting Multimedia Objects in Peer-to-Peer Overlay Networks. *Proc. of IEEE 19th Conference on Advanced Information Networking and Applications(AINA-2005)*, 1:165–170, 2005.

[14] K. Watanabe, N. Hayashibara, T. Enokido, and M. Takizawa. CBF: Look-up protocol for distributed multimedia objects in peer-to-peer overlay networks. *Journal of Interconnection Networks (JOIN)*, 6(3):323–344, 2005.

[15] K. Watanabe and M. Takizawa. Service oriented cooperation among trustworthy peers. *accepted at JOIN*, 2006.

[16] L. Xiong and L. Liu. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.