Ranking Factors in Peer-to-Peer Overlay Networks

KENICHI WATANABE and YOSHIO NAKAJIMA
Tokyo Denki University
TOMOYA ENOKIDO
Rissho University
and
MAKOTO TAKIZAWA
Tokyo Denki University

A large number of peer processes are distributed in a peer-to-peer (P2P) overlay network. It is difficult, maybe impossible for a peer to perceive the membership and location of every resource object due to the scalability and openness of a P2P network. In this article, we discuss a fully distributed P2P system where there is no centralized controller. Each peer has to obtain service information from its acquaintance peers and also send its service information to the acquaintance peers. An acquaintance peer of a peer p is a peer about whose service the peer p knows and with which the peer p can directly communicate in an overlay network. Some acquaintance peer might hold obsolete service information and might be faulty. Each peer has to find a more trustworthy one among acquaintance peers. There are many discussions on how to detect peers that hold a target object. However, a peer cannot manipulate an object without being granted access rights (permissions). In addition to detecting what peers hold a target object, we have to find peers granted access rights to manipulate the target object. The trustworthiness of each acquaintance is defined in terms of the satisfiability and ranking factor in this article. The satisfiability of an acquaintance peer shows how much each peer can trust the acquaintance peer through direct communication to not only detect target objects but also obtain their access rights. On the other hand, the ranking factor of an acquaintance peer indicates how much the acquaintance peer is trusted only by trustworthy acquaintance peers which is different from the traditional reputation concept. We evaluate how

This article is based on the paper "Ttrustworthiness in Peer-to-Peer Overlay Networks" By Kenichi Watanabe, Yoshio Nakajima, Tomoya Enokido, and Makoto Takizawa which appears in Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 86–93. © 2006 IEEE.
This research is partially supported by the Research Institute for Science and Technology (Q06J-07) and the Frontier Research and Development Center (18-J-6), Tokyo Denki University.
Authors' addresses: K. Watanabe, Y. Nakajima, and M. Takizawa, Distributed Systems Laboratory, Tokyo Denki University, Ishizaka, Hatoyama, Hiki, Saitama 350-0394, Japan; email: nabe@takilab.k.dendai.ac.jp; T. Enokido, Faculty of Business Administration, Rissho University, Osaki, Shinagawa-ku, Tokyo 141-8602, Japan.

the trustworthiness of an acquaintance peer is changed through interactions among peers in a detection algorithm.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms: Reliability, Security

Additional Key Words and Phrases: P2P overlay networks, acquaintances, ranking factor, satisfiability, trustworthiness

ACM Reference Format:

Watanabe, K., Nakajima, Y., Enokido, T., and Takizawa, M. 2007. Ranking factors in peer-to-peer overlay networks. ACM Trans. Autonom. Adapt. Syst. 2, 3, Article 11 (September 2007), 26 pages. DOI = 10.1145/1278460.1278465 http://doi.acm.org/10.1145/1278460.1278465

1. INTRODUCTION

Various types and a huge number of peer computers are interconnected and the membership is dynamically changed in a peer-to-peer (P2P) overlay network [Liu et al. 2004; Androutsellis-Theotokis and Spinellis 2004]. A group of multiple peer processes (abbreviated peers) on peer computers cooperate to achieve some objectives by manipulating objects and exchanging messages in an overlay network. An object is a unit of resource like a tuple and table in a relational database [Oracle Corporation 1999; SYBASE SQL Server]. An object is an encapsulation of data and methods for manipulating the data [OMG Inc. 1997]. Service supported by each object is characterized by types of methods. An object is distributed to peers in various ways like downloading and caching [Watanabe et al. 2005a, 2005b] in P2P overlay networks. In this article, we consider a fully distributed P2P model where there is neither centralized index nor coordination. Every peer is autonomous and can obtain information on service through communication with other peers.

Peers are classified into *object holder peers*, *permission holder peers*, and *intermediate peers* according to the types of service which are supported by the peers. An object holder peer of an object is a peer which holds the object. A permission holder peer of an object is a peer which is granted access rights, that is, permission on the object. An intermediate peer is a peer that can help other peers satisfy their requirements. For example, an intermediate peer can manipulate objects in remote peers on behalf of another peer. Here an *acquaintance* peer of a peer p is a peer with which the peer p can directly communicate and whose service the peer p knows [Watanabe et al. 2005a, 2005b; Watanabe and Takizawa 2006; Watanabe et al. 2006]. A peer p first asks its acquaintance peers to detect target peers that can manipulate a target object so as to satisfy an access request which the peer p issues. Even if some peer holds a target object, that is, object holder peer, the peer cannot be asked to manipulate the object if the peer is not granted an access right (permission) on the object. If acquaintance peers which satisfy the access request are not detected, each acquaintance peer furthermore asks its acquaintance peers. Thus, access requests are propagated from acquaintance peers to acquaintance peers. Acquaintance concepts have so far been discussed only to detect target peers holding target objects [Crespo and Garcia-Molina 2002; Egemen et al. 2005]. In the papers

Watanabe and Takizawa [2006] and Watanabe et al. [2006], the authors discussed how peers cooperate with each other to obtain a required service, for example, find a permission peer that can manipulate a target object and then ask the permission peer to manipulate the object.

If service supported by a peer is changed, the change information is propagated through acquaintance peers. However, it takes time to propagate the change of the service to every peer due to the scalability and openness of a P2P overlay network. Hence, some acquaintance peers of a peer may show obsolete and inconsistent information on target peers of a target object. In addition, acquaintance peers may not only stop by fault but may also be arbitrarily faulty [Lamport et al. 1982]. Hence, it is critical to discuss how much a peer trusts its acquaintance peer. A requesting peer p is satisfiable for each access request to find a target peer if a target peer is detected. However, if the requesting peer p is not granted an access right, the peer p is not satisfiable to manipulate a target object even if the peer p finds where the target object exists. We define the *satisfiability* σ_{ij} of a peer p_i to an acquaintance peer p_j with respect to a type of access requests, that is, to find an object, to manipulate an object, and to grant an access right of an object. The more satisfiable that the replies an acquaintance peer p_j returns to a requesting peer p_i are, the more trustworthy the acquaintance peer p_j is for the requesting peer p_i. Thus, we define the *trustworthiness* τ_{ij} of a requesting peer p_i to an acquaintance peer p_j by aggregating the satisfiability of each access request obtained through each interaction with the acquaintance peer p_j. An acquaintance peer p_j may introduce its acquaintance peer p_k to a requesting peer p_i. If the peer p_k returns a more satisfiability reply to the requesting peer p_i, the peer p_i not only trusts the peer p_k more, but also the acquaintance peer p_j. Thus, the trustworthiness to the acquaintance peer p_j is also changed on receipt of a reply from the peer p_k. Next, we define the *ranking factor* ρ_{ij} of a peer p_i to an acquaintance peer p_j by showing how much the acquaintance peer p_j is trusted. The traditional reputation concept [Kamvar et al. 2003; Cuenca-Acuna et al. 2002] implies how much a peer is trusted by other peers. We discuss two types of ranking factors with respect to each type of access requests. First, we define the ranking factor ρ_{ij} of a peer p_i to an acquaintance peer p_j to show how much the acquaintance p_j is trusted only by other acquaintance peers of the peer p_i which are trustworthy to the peer p_i. Only the acquaintances of a requesting peer p_i are taken into account and less trustworthy acquaintance peers are neglected in the requesting peer p_i. In real life, each person p_i asks his/her friends about how much some person p_j can be trusted. If an opinion of some friend p_k about the person p_j is quite different from his/her own opinion, the person p_i does not listen to p_k's opinion. In the second type of the ranking factor, only the trustworthy acquaintance peers showing trustworthiness to p_j which are not very different from the trustworthiness of p_i to p_j are considered. In the paper Watanabe and Takizawa [2006], we discussed how a peer p_i and acquaintance peers of the peer p_i cooperate with each other to obtain satisfiable replies for each type of access requests.

The acquaintance relations are propagated through P2P interactions in a P2P overlay network. Each peer p_i can admit only a limited amount of the acquaintance relations in its acquaintance base AB_i. Obsolete and untrustworthy

acquaintance relations are thrown away to make space to store new acquaintance relations if the acquaintance base is full. We implement the flooding algorithm for detecting target peers where each request is forwarded to more trustworthy acquaintances. A peer sends a request reply message with trustworthiness and ranking factor information to an acquaintance peer. On receipt of a message from an acquaintance, a peer p_i updates its acquaintance base AB_i. We evaluate the flooding algorithm on peers in terms of hit ratio and the number of messages and how the satisfiability of each peer is changed through communication with acquaintance peers.

The rest of this article is organized as follows. In Section 2, we present acquaintance relations of peers. In Section 3, we discuss the trustworthiness and ranking factors of an acquaintance peer. In Section 4, we discuss how to implement peers. In Section 5, we evaluate the detection algorithm, that is, how each peer can detect target peers in cooperation with acquaintance peers.

2. ACQUAINTANCE PEERS

2.1 Peer-to-Object (P2O) Relations

In peer-to-peer (P2P) overlay networks [Clarke et al. 2000; Cuenca-Acuna et al. 2002; Napster ; Ratnasamy et al. 2001; Ripeanu 2001; Rowstron and Druschel 2001; Stoica et al. 2003; Zhao et al. 2001], only how to detect a target peer which holds a target object is discussed. Even if the location of a target object is detected in a P2P overlay network, the target object cannot be manipulated without being granted an access right (permission). Thus, a peer is required to be granted an access right $[o, op]$ to manipulate an object o in a method op in addition to discovering in which peers the object exists. A peer is *authorized* for an object if the peer is granted an access right on the object. Only a peer granted the access right is allowed to manipulate the object. Hence, we discuss relations among peers and objects by taking into account the authorization of access rights.

We have to find target peers which support satisfiable service on a target object and which are allowed to manipulate the object in a P2P overlay network. An object o can be manipulated only through a method op. First, an application issues an access request $\langle o, op \rangle$ to a local peer p to manipulate a target object o with a method op. Here the requesting peer p is referred to as an *initial* requesting peer of the access request $\langle o, op \rangle$. A *target* peer of an access request $\langle o, op \rangle$ is a peer that can manipulate a target object o through a required method op. An object is replicated in multiple peers. For example, an object may be downloaded to peers and the peers hold replicas of the object. Hence, there might be multiple target peers of an access request $\langle o, op \rangle$ which can manipulate replicas of the object o through a method op.

On receipt of an access request $\langle o, op \rangle$ from a requesting peer, a peer p has to find target peers of the access request. It is difficult, maybe impossible, for each peer to perceive which service of which objects every other peer supports due to the scalability and openness of a P2P overlay network. If the peer p can not manipulate the object o, the peer p forwards the access request $\langle o, op \rangle$ to

another acquaintance peer p'. Here a pair of the peers p and p' are referred to as *requesting* and *requested* peers of $\langle o, op \rangle$, respectively, [Nakajima et al. 2006].

Let $\mathbf{P}$ be a set of peers and $\mathbf{O}$ be a set of objects in a P2P overlay network. There are the following types of peer-to-object (P2O) relations, $|$, $\models$, $\stackrel{s}{\rightarrow}$, $\vdash$, $\Box$, and $\triangleright$ ($\subseteq \mathbf{P} \times \mathbf{O}$) for a peer p, an object o, and a method op.

[P2O relations 1]

(1) *Object holder* peer: a peer p which *holds* an object o ($p \mid o$) if the object o is stored in the peer p.

(2) *Permission holder* peer: a peer p which is granted some access right. There are two types of permission holder peers.
   (a) *Manipulation* peer: a peer p which can *manipulate* an object o through a method op ($p \models_{op} o$), that is, the peer p is granted an access right $[o, op]$.
   (b) *Authorized* peer: a peer p which *can grant* an access right $[o, op]$ to another peer ($p \vdash_{op} o$).

(3) *Intermediate* peer: a peer p which knows information on a target object o and can help a requesting peer satisfy its requirements ($p \rightarrow_{op} o$). There are two types of intermediate peers.
   (a) *Surrogate* peer: a peer p which can satisfy requirements of requesting peers on behalf of the peers ($p \stackrel{s}{\rightarrow}_{op} o$).
   (b) *Informing* peer: a peer p which can inform a requesting peer of information on a target object ($p \stackrel{i}{\rightarrow}_{op} o$).

(4) *Independent* peer: a peer p which is not only an object holder peer but also a permission holder peer ($p \triangleright_{op} o$). Otherwise, the peer p is referred to as a *dependent* peer ($p \not\triangleright o$).

(5) *Serving* peer: a peer p which can do something for an object o by using a method op ($p \Box_{op} o$) if and only if the peer p is an object holder peer, a permission holder peer, or an intermediate peer.

Even if a peer p holds an object o ($p \mid o$), the peer p may not be granted an access right $[o, op]$ on the object o ($p \not\models_{op} o$). If a peer p can grant an access right $[o, op]$ to another peer ($p \vdash_{op} o$), the peer p is granted the access right $[o, op]$ ($p \models_{op} o$). In the discretionary access control (DAC) model [Ferraiolo et al. 2003; Oracle Corporation 1999; SYBASE SQL Server], a peer p can grant an access right $[o, op]$ if the peer p is granted the access right $[o, op]$, that is, $p \vdash_{op} o$ if $p \models_{op} o$. For example, a peer p_1 can read and write a file f, that is, a pair of access rights $[f, read]$ and $[f, write]$ are granted to the peer p_1. The peer p_1 can grant the access right $[f, read]$ to another peer p_2. The peer p_2 can further grant the access right $[f, read]$ to another peer p_3. On the other hand, in the mandatory access control (MAC) model [Oracle Corporation 1999; SYBASE SQL Server], a peer p cannot grant an access right to another peer even if the peer p is granted the access right. Only the centralized authorized peer of a target object o, for example, owner of an object, can grant an access right $[o, op]$ to other peers. If a peer would like to obtain an access right, the peer has to ask the centralized authorized peer to grant the access right.

If a peer p is a surrogate peer ($p \stackrel{s}{\rightarrow}_{op} o$), p is a manipulation peer ($p \models_{op} o$). A manipulation peer p might not manipulate an object o even if a requesting peer asks the peer p to manipulate the object o. Only if the peer p would like to manipulate the object o through the method op for the requesting peer, the peer p manipulates the object o. If a requesting peer p' asks a surrogate peer p to manipulate an object o in a method op, the surrogate peer p manipulates the object o on behalf of the requesting peer p'.

The following types of P2O relations are defined from the relations $\models_{op}$, $\stackrel{s}{\rightarrow}_{op}$, and $\vdash_{op}$ with a method op and an object o.

[P2O relations 2]

—A peer p can manipulate an object o ($p \models o$) if $p \models_{op} o$ for some method op.

—A peer p can manipulate an object o on behalf of another peer ($p \stackrel{s}{\rightarrow} o$) if $p \stackrel{s}{\rightarrow}_{op} o$ for some method op.

—A peer p can grant an access right of an object o to another peer ($p \vdash o$) if $p \vdash_{op} o$ for some method op.

—A peer p can directly manipulate an object o ($p \triangleright o$) iff $p \triangleright_{op} o$ for some method op.

—A peer p can do something for an object o ($p \Box o$) iff $p \Box_{op} o$ for some method op.

2.2 Acquaintance (P2P) Relations

Each peer cannot perceive in which peers each object exists and how each object can be manipulated due to the scalability of a peer-to-peer (P2P) overlay network. Each peer obtains service information on objects from other peers. We discuss acquaintance relations among peers by using the peer-to-object (P2O) relations $|$, $\models$, $\stackrel{s}{\rightarrow}$, and $\vdash$ discussed in the preceding section. *Acquaintance* peers of a peer p are peers whose services the peer p knows, that is, object holder, manipulation, surrogate, and authorized peers. For example, if a peer p knows that another peer p_i can manipulate an object o in a method op ($p_i \models_{op} o$), the peer p_i is an acquaintance peer of the peer p. If a peer p knows that another peer p_i has an acquaintance peer p_j, the peer p_i is also an acquaintance peer of the peer p. Acquaintance information of a peer p is stored in an *acquaintance base* of the peer p.

We discuss what kinds of acquaintance relations among peers there are. An *acquaintance* relation $\rightarrow$ is formally defined as a relation $\rightarrow \subseteq \mathbf{P} \times 2^{\mathbf{P} \times \mathbf{O}}$. For a peer $p_i \in \mathbf{P}$, an acquaintance relation "$p_i \rightarrow \mathbf{PO}_i$" holds if a peer p_i perceives a P2O relation $\mathbf{PO}_i \subseteq \mathbf{P} \times \mathbf{O}$. Here $p_i \rightarrow (p_j \Box o)$ if $p_i \rightarrow \mathbf{PO}_i$ and $p_j \Box o \in \mathbf{PO}_i$, that is, a peer p_i perceives that another peer p_j is in a P2O relation $\Box$ with an object o. There are the following types of acquaintance relations for a peer p, an object o, and a method op.

(1) A relation "$p \rightarrow (p_i \mid o)$" holds iff a peer p perceives that another peer p_i holds an object o ($p_i \mid o$). That is, a peer p knows that a peer p_i is an object holder peer of an object o with respect to a method op. The peer p_i is an *object holder acquaintance* of the peer p with respect to an object o.

(2) $p \rightarrow (p_i \models_{op} o)$ iff a peer p perceives that a peer p_i can manipulate an object o through a method op ($p_i \models_{op} o$). That is, a peer p knows that another peer p_i is a manipulation peer of an object o by a method op. The peer p_i is a *manipulation acquaintance* peer of the peer p with respect to an access request $\langle o, op \rangle$.

(3) $p \rightarrow (p_i \stackrel{s}{\rightarrow}_{op} o)$ iff a peer p perceives that a peer p_i is a surrogate peer of an access request $\langle o, op \rangle$. The peer p_i is a *surrogate acquaintance* peer of the peer p with respect to an access request $\langle o, op \rangle$.

(4) $p \rightarrow (p_i \vdash_{op} o)$ iff a peer p perceives that a peer p_i can grant an access right $[o, op]$ ($p_i \vdash_{op} o$). The peer p_i is an *authorized acquaintance* peer of the peer p with respect to an access request $\langle o, op \rangle$.

A peer p can issue an access request $\langle o, op \rangle$ to an object holder peer p_i of an object o if $p \rightarrow (p_i \mid o)$ and $p \models_{op} o$. Suppose the peer p is granted the access right $[o, op]$ and knows that another peer p_i holds the object o. The peer p can issue the access request $\langle o, op \rangle$ to the object o in the peer p_i.

The following acquaintance relations with a peer p are defined for an object o and a peer p_i.

—$p \rightarrow (p_i \models o)$ if $p \rightarrow (p_i \models_{op} o)$ for some method op.
—$p \rightarrow (p_i \stackrel{s}{\rightarrow} o)$ if $p \rightarrow (p_i \stackrel{s}{\rightarrow}_{op} o)$ for some method op.
—$p \rightarrow (p_i \vdash o)$ if $p \rightarrow (p_i \vdash_{op} o)$ for some method op.

For a P2O relation $\square \in \{ \mid, \models, \stackrel{s}{\rightarrow}, \vdash \}$, the following relations are defined for a pair of peers p and p_i, an object o, and a method op.

—$p \rightarrow (p_i \square_{op} o)$ iff $p \rightarrow (p_i | o)$, $p \rightarrow (p_i \models_{op} o)$, $p \rightarrow (p_i \stackrel{s}{\rightarrow}_{op} o)$, or $p \rightarrow (p_i \vdash_{op} o)$.
—$p \rightarrow^* (p_i \square_{op} o)$ iff $p \rightarrow (p_i \square_{op} o)$ or $p \rightarrow (p_k \rightarrow^* (p_i \square_{op} o))$ for some peer p_k where $\square \in \{ \mid, \models, \stackrel{s}{\rightarrow}, \vdash \}$.
—$p \rightarrow^+ (p_i \square_{op} o)$ iff $p \rightarrow (p_k \rightarrow^* (p_i \square_{op} o))$ for some peer p_k.
—$p \rightarrow (p_i \square o)$ iff $p \rightarrow (p_i \square_{op} o)$ for some method op.
—$p \rightarrow^* (p_i \square o)$ iff $p \rightarrow^* (p_i \square_{op} o)$ for some method op.
—$p \rightarrow^+ (p_i \square o)$ iff $p \rightarrow^+ (p_i \square_{op} o)$ for some method op.

An acquaintance peer of a peer p_i is another peer p_j which knows where objects are held, how objects can be manipulated, and what access rights the peer p_j can be granted. The following types of acquaintance relations $\Rightarrow_o^{\square_{op}}$, $\Rightarrow_o^{\square}$, $\Rightarrow_o$, and $\Rightarrow$ ($\subseteq \mathbf{P} \times \mathbf{P}$) are defined for a set $\mathbf{P}$ of peers [Nakajima et al. 2006].

[Acquaintance relations]

—A peer p_j is an acquaintance peer of a peer p_i with respect to an access request $\langle o, op \rangle$ and a P2O relation $\square$ ($\in \{ \mid, \models, \stackrel{s}{\rightarrow}, \vdash \}$) ($p_i \Rightarrow_o^{\square_{op}} p_j$) if one of the following conditions holds.

—$p_i \rightarrow (p_j \ \Box_{op} \ o)$.
—The peer p_i perceives "$p_k \rightarrow (p_j \ \Box_{op} \ o)$" for some peer p_k, that is, $p_i \rightarrow (p_k \rightarrow (p_j \Box_{op} o))$.
—p_i perceives "$p_k \Rightarrow_o^{\Box_{op}} p_j$" for some peer p_k, that is, $p_i \rightarrow (p_k \Rightarrow_o^{\Box_{op}} p_j)$.

—$p_i \Rightarrow_o^{\Box} p_j$ iff $p_i \Rightarrow_o^{\Box_{op}} p_j$ for some method op.
—A peer p_j is an acquaintance peer of a peer p_i on an object o with respect to a method op ($p_i \Rightarrow_o^{op} p_j$) if $p_i \Rightarrow_o^{\Box_{op}} p_j$ for some P2O relation $\Box$.
—$p_i \Rightarrow_o p_j$ iff $p_i \Rightarrow_o^{op} p_j$ for some method op.
—A peer p_j is an acquaintance peer of a peer p_i ($p_i \Rightarrow p_j$) if $p_i \Rightarrow_o p_j$ for some object o.

Following the P2O relations, we further define the following types of P2O relations $\Box^*$ and $\Box^+$ ($\subseteq \ \mathbf{P} \times \ \mathbf{O}$).

—$p \ \Box_{op}^* \ o$ iff $p \ \Box_{op} \ o$ or $p \Rightarrow_o^{\Box_{op}} p_i$ for some peer p_i, that is, a peer p directly or indirectly makes an access to an object o by a method op.
—$p \ \Box^* \ o$ iff $p \ \Box_{op}^* \ o$ for some method op.
—$p \ \Box_{op}^+ \ o$ iff $p \Rightarrow_o^{\Box_{op}} p_i$ for some peer p_i, that is, a peer p indirectly makes an access to an object o through a method op.
—$p \ \Box^+ \ o$ iff $p \ \Box_{op}^+ \ o$ for some method op.

If $p_i \Rightarrow_o^{\mid} p_j$, a peer p_j is referred to as an object holder acquaintance peer of an object o. If $p_i \Rightarrow_o^{\models} p_j$, $p_i \Rightarrow_o^{\overrightarrow{s}} p_j$, and $p_i \Rightarrow_o^{\vdash} p_j$, a peer p_j is a manipulation, surrogate, and authorized acquaintance peer of an object o, respectively. If $p_i \Rightarrow_o^{\Box} p_j$, $p_i \Rightarrow_o^{\Box} p_k$, $p_j \Rightarrow_o^{\Box} p_k$, and $p_j \not\Box o$ ($p_j \ \Box \ o$ does not hold), a peer p_j is referred to as a *closer* acquaintance peer of a peer p_i than another peer p_k with respect to an object o.

Last, the acquaintance relation $\Rightarrow$ between a pair of peers p_i and p_j is defined as follows.

—$p_i \Rightarrow p_j$ (a peer p_j is an acquaintance peer of a peer p_i) iff $p_i \Rightarrow_o p_j$ for some object o.

The acquaintance relation of peers is reflexive but is neither symmetric nor transitive. For example, even if a peer p_j is an acquaintance peer of a peer p_i ($p_i \Rightarrow p_j$), an acquaintance relation $p_j \Rightarrow p_i$ may not hold. Even if $p_i \Rightarrow p_j$ and $p_j \Rightarrow p_h$, $p_i \Rightarrow p_h$ may not hold. If $p_i \Rightarrow p_j$ and $p_j \Rightarrow p_i$, a pair of peers p_i and p_j are referred to as *friend* peers ($p_i \Leftrightarrow p_j$).

Let $view(p_i)$ be a set $\{p_j \mid p_i \Rightarrow p_j\}$ of acquaintance peers of a peer p_i. A set $view(p_i)$ is referred to as *view* of a peer p_i. Let $cview(p_i)$ be a set $\{p_j \mid p_i \Rightarrow p_j$ but there is no peer p_k such that $p_i \Rightarrow p_k \Rightarrow p_j\}$ of closest acquaintance peers of a peer p_i. The view $view(p_i)$ is maintained by exchanging information with the acquaintance peers as discussed in the paper Watanabe et al. [2005a].

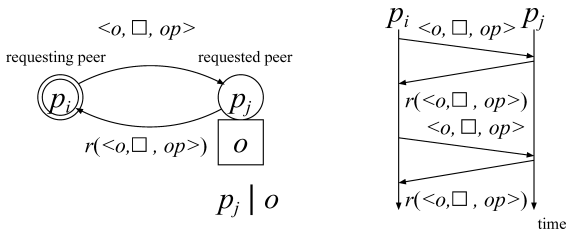

Fig. 1. Interaction with an acquaintance peer.

3. TRUSTWORTHINESS OF ACQUAINTANCE PEERS

3.1 Satisfiability of an Access Request

Each requesting peer p_i has to find acquaintance peers to which an access request $\langle o, \square, op\rangle$ can be issued, where an object o is an object, op is a method, and $\square$ is a peer-to-object (P2O) relation. Each peer p_i has a view $view(p_i)$ which is a set of its acquaintance peers. The requesting peer p_i tries to find the most trustworthy peer among the acquaintance peers in $view(p_i)$ with respect to the access request $\langle o, \square, op\rangle$. The peer p_i issues an access request $\langle o, \square, op\rangle$ to some acquaintance peer p_j in the view $view(p_i)$ to manipulate a target object o in a method op. If the requested acquaintance peer p_j immediately obtains the reply $r(\langle o, \square, op\rangle)$ of the access request $\langle o, \square, op\rangle$ and sends the reply $r(\langle o, \square, op\rangle)$ to the requesting peer p_i and the reply of the access request is satisfiable, the peer p_i considers the acquaintance peer p_j is satisfiable for the access request $\langle o, \square, op\rangle$ (Figure 1). On the other hand, if it takes a longer time to obtain the reply of the access request $\langle o, \square, op\rangle$ or the reply is not satisfiable, the acquaintance peer p_j is less satisfiable for the requesting peer p_i. The requesting peer p_i can trust an acquaintance peer p_j more if the acquaintance peer p_j returns a satisfiable reply to the requesting peer p_i.

We define the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$ of a requesting peer p_i to an acquaintance peer p_j with respect to an access request $\langle o, \square, op\rangle$ in terms of services supported by the requesting peer p_i and acquaintance peer p_j. Service supported by each peer p_i is characterized in terms of a P2O relation $p_i \square o$ and acquaintance relation $p_i \rightarrow (p_j \square_{op} o)$. For example, the state of a requesting peer p_i is $p_i \mid o$ if the peer p_i holds an object o. $p_j \rightarrow (p_h \mid o)$ indicates that an acquaintance peer p_j perceives its acquaintance peer p_h to be an object holder peer of an object o. Table I summarizes the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$ for an access request $\langle o, \square, op\rangle$ issued to an acquaintance peer p_j by a peer p_i. In the table, services of peers p_i and p_j show services supported by the requesting peer p_i and the requested acquaintance peer p_j, respectively.

Suppose a requesting peer p_i issues an access request $\langle o, |, \_\rangle$ to an acquaintance peer p_j to find an object holder peer of a target object o. If the acquaintance peer p_j holds the object o ($p_j \mid o$), the acquaintance peer p_j sends a reply $r(\langle o, |, \_\rangle)$ with a positive acknowledgment to the requesting peer p_i (Figure 2). Here the requesting peer p_i finds the acquaintance peer p_j to hold an object o. The satisfiability $\sigma_{ij}(\langle o, |, \_\rangle)$ is 1, that is, the requesting peer p_i is satisfied for the access request $\langle o, |, \_\rangle$ to find an object holder peer of the object o since the

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

States of p_i	Access Requests	States of Acquaintance Peer p_j	Satisfiability σ_{ij}
$p_i \mid o$ and $p_i \models_{op} o$	$\langle o, \mid, - \rangle$	-	$\sigma_{ii} = 1$
$p_i \models_{op} o$ and $p_i \not\parallel o$		$p_j \mid o$	$\sigma_{ij} = 1$
		$p_j \rightarrow (p_k \mid o)$	$\sigma_{ij} = \delta_i, \sigma_{ik} = 1$
$p_i \mid o$ 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 $p_i \not\parallel o$	$\langle o, \mid, - \rangle$	$p_j \mid o$	$\sigma_{ij} = 1$
		$p_j \rightarrow (p_k \mid o)$	$\sigma_{ij} = \delta_i, \sigma_{ik} = 1$
$p_i \not\Box_{op} o$	$\langle o, \models, op \rangle$	$p_j \models_{op} o$	$\sigma_{ij} = 1$
	$\langle o, \vdash, op \rangle, \langle o, \mid, - \rangle$	$p_j \vdash_{op} o$, $p_k \mid o$	$\sigma_{ij} = \delta_i, \sigma_{ik} = \delta_i$
	$\langle o, \models, op \rangle, \langle o, \mid, - \rangle$	$p_j \models_{op} o$, $p_k \mid o$	$\sigma_{ij} = \delta_i, \sigma_{ik} = \delta_i$

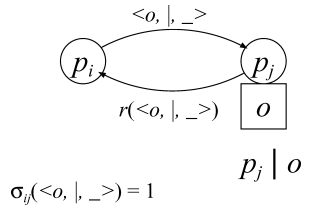

Fig. 2. Detection request.

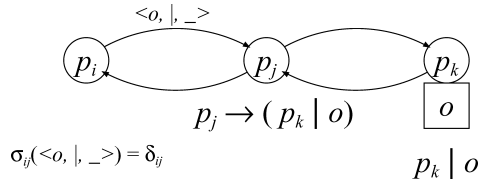

Fig. 3. Detection request.

peer p_i can directly obtain the reply $r(\langle o, |, \_\rangle)$ of the access request $\langle o, |, \_\rangle$ from the acquaintance peer p_j.

Next, if the acquaintance peer p_j is not an object holder peer ($p_j \not\parallel o$) but the peer p_j knows another peer p_k is an object holder peer of the object o ($p_j \rightarrow (p_k \mid o)$), the requesting peer p_i cannot get the reply from the acquaintance peer p_j but may get the reply from the other peer p_k (Figure 3). The requesting peer p_i is less satisfiable since the peer p_i cannot directly obtain service from the acquaintance peer p_j, for example, it might take a longer time to forward an access request to the object holder peer p_k. The satisfiability $\sigma_{ij}(\langle o, |, \_\rangle)$ is defined to be δ_{ij} (≤ 1). Here δ_{ij} is referred to as a *distance factor* between a pair of peers p_i and p_j. If a peer p_l is an acquaintance peer of the peer p_i and $p_l \rightarrow (p_m \rightarrow (p_k \mid o))$, the satisfiability $\sigma_{il}(\langle o, |, -\rangle)$ is defined to be $\delta_{il} \cdot \sigma_{lk}(\langle o, |, -\rangle) = \delta_{il} \cdot \delta_{lm} \cdot \sigma_{mk}(\langle o, |, -\rangle)$. We postulate that the more the number of peers an access request passes, the less satisfiable the requesting peer is in this article.

For an access request $\langle o, \models, op\rangle$ to manipulate an object o in a method op, if a requesting peer p_i is granted an access right $[o, op]$ ($p_i \models_{op} o$) and knows

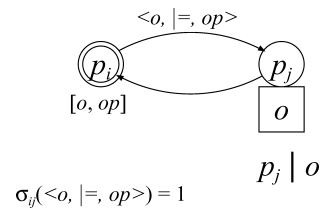

Fig. 4. Manipulation request.

that another peer p_j holds the object o ($p_i \rightarrow (p_j \mid o)$), the requesting peer p_i obtains the reply $r(\langle o, \models, op\rangle)$ by issuing the method op to the object o in the object holder peer p_j (Figure 4). Hence the satisfiability $\sigma_{ij}(\langle o, \models, op\rangle)$ is 1.

3.2 Trustworthiness

A requesting peer p_i obtains the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$ to an acquaintance peer p_j each time the peer p_i issues an access request $\langle o, \square, op\rangle$ to the acquaintance peer p_j. The peer p_i trusts the acquaintance peer p_j more if the peer p_i receives more satisfiable replies from the acquaintance peer p_j as discussed. The trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ of a requesting peer p_i to an acquaintance peer p_j with respect to an access request $\langle o, \square, op\rangle$ is obtained by aggregating the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$ obtained each time an access request $\langle o, \square, op\rangle$ is issued to the peer p_j. For simplicity, a pair of notations τ_{ij} and σ_{ij} stand for the trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ and the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$, respectively. The satisfiability σ_{ij} obtained at each interaction with an acquaintance peer p_j is kept on record by the requesting peer p_i. The trustworthiness τ_{ij} is calculated by the following function $Trust0$ for the current trustworthiness τ_{ij} and the satisfiability σ_{ij} which was just obtained.

$$Trust0(\tau_{ij}, \sigma_{ij}, \alpha_i) := \alpha_i \cdot \tau_{ij} + (1 - \alpha_i) \cdot \sigma_{ij}. \tag{1}$$

Suppose a peer p_i obtains the satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$ by issuing an access request $\langle o, \square, op\rangle$ to an acquaintance peer p_j and then receiving a reply $r(\langle o, \square, op\rangle)$ from the acquaintance peer p_j. The trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ is changed with the function $Trust0(\tau_{ij}, \sigma_{ij}, \alpha_i)$. Initially, the trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ 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 τ_{ij} is not changed even if the current satisfiability σ_{ij} is obtained and σ_{ij} is quite different from the previous ones. If $\alpha_i = 0$, the trustworthiness τ_{ij} is decided only by the current satisfiability σ_{ij}. The smaller the constant α_i is, the more important the satisfiability σ_{ij} obtained for a current access request $\langle o, \square, op\rangle$ is. For example, let a peer p_j be an acquaintance peer of a requesting peer p_i. After the peer p_i sends access request messages of $\langle o, \square, op\rangle$ to the acquaintance peer p_j, the requesting peer p_i obtains the trustworthiness τ_{ij} to the peer p_j. Suppose $\alpha_i = 0.9$ and $\tau_{ij} = 0.5$. The peer p_i newly issues an access request message $\langle o, \square, op\rangle$ to the acquaintance peer p_j and receives the reply $r(\langle o, \square, op\rangle)$ whose satisfiability $\sigma_{ij}(\langle o, \square, op\rangle)$ is 0.7. The new trustworthiness τ_{ij} is calculated as follows. $Trust0(\tau_{ij}, \sigma_{ij}, \alpha_i) = 0.9 \cdot 0.5 + (1 - 0.9) \cdot 0.7 = 0.45 + 0.07 = 0.52$. Since the satisfiability σ_{ij} is 0.7, that is, larger than $\tau_{ij} = 0.5$, the trustworthiness τ_{ij} is increased to 0.52 from 0.5.

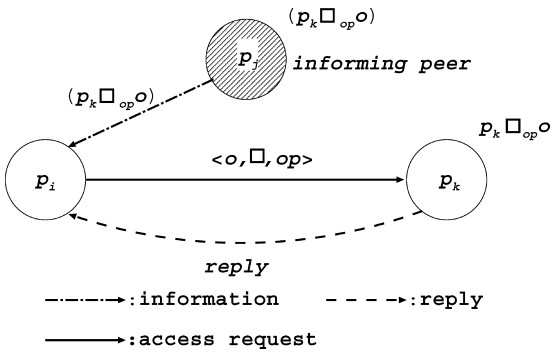

Fig. 5. Introduction of peer.

Next, suppose a peer p_i issues an access request $\langle o, \square, op\rangle$ to an acquaintance peer p_j. The acquaintance peer p_j does not support the P2O relation $p_j \square_{op} o$ but perceives that some peer p_k supports the required service, that is, $p_j \not\square_{op} o$ but $p_j \xrightarrow{i} (p_k \square_{op} o)$. On receipt of the access request $\langle o, \square, op\rangle$ from the requesting peer p_i, the acquaintance peer p_j informs the requesting peer p_i of the P2O relation $p_k \square_{op} o$, that is, p_k is a serving peer of the access request $\langle o, \square, op\rangle$. Here the acquaintance peer p_j is an informing peer of the serving peer p_k with the access request $\langle o, \square, op\rangle$. There are two choices, the requesting peer p_i directly manipulates the object o in the serving peer p_k or asks the acquaintance peer p_j to manipulate the object o in the serving p_k. Suppose the requesting peer p_i directly issues an access request $\langle o, \square, op\rangle$ to the serving peer p_k (Figure 5). If the peer p_i receives the reply $r(\langle o, \square, op\rangle)$ from the serving peer p_k, the satisfiability $\sigma_{ik}(\langle o, \square, op\rangle)$ to the peer p_k is obtained according to Table I. The trustworthiness $\tau_{ik}(\langle o, \square, op\rangle)$ of the requesting peer p_i to the serving peer p_k is obtained by the function *Trust*0 as discussed. The trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ of the requesting peer p_i to the informing peer p_j is also changed by the following function for the current trustworthiness τ_{ij}, the satisfiability σ_{ik}, the trustworthiness τ_{ik} which were just obtained from the requested peer p_k, the constant α_i, and the constant β_i.

$$Trust1(\tau_{ij}, \sigma_{ik}, \tau_{ik}, \alpha_i, \beta_i) := (1 - \beta_i) \cdot \tau_{ij} + \beta_i \cdot \tau_{ij} \cdot Trust0(\tau_{ik}, \sigma_{ik}, \alpha_i)/\tau_{ik}. \quad (2)$$

Here if $\tau_{ij} \cdot Trust0(\tau_{ik}, \sigma, \alpha)/\tau_{ik} > 1$, it is normalized to be 1. β_i is the inverse of the number of hops from the informing peer p_j to the serving peer p_k and $0 < \beta_i \leq 1$. For example, let us consider Figure 5 where p_i is a requesting peer, p_j shows an informing peer, and p_k indicates a serving peer of an access request $\langle o, \square, op\rangle$. Suppose $\tau_{ij} = 0.5$, $\tau_{ik} = 0.4$, $\sigma_{ik} = 0.8$, and $\alpha_i = 0.9$. Since the number of hops between a pair of the requesting peer p_j and the serving peer p_k introduced by the acquaintance peer p_j is 1, $\beta_i = 1 / 1 = 1$. The new trustworthiness τ_{ij} is calculated. $Trust1(\tau_{ij}, \sigma_{ik}, \tau_{ik}, \alpha_i, \beta_i) = (1 - 1) \cdot 0.5 + 1 \cdot 0.5 \cdot \{0.9 \cdot 0.4 + (1 - 0.9) \cdot 0.8\} / 0.4 = 0.5 \cdot 0.44 / 0.4 = 0.55$. Since the informing peer p_j introduces a more trustworthy serving peer p_k to the requesting peer p_i, the trustworthiness $\tau_{ij}(\langle o, \square, op\rangle)$ of the requesting peer p_i to the informing peer p_j is increased to 0.55 from 0.5. Next, suppose the requesting peer p_i does

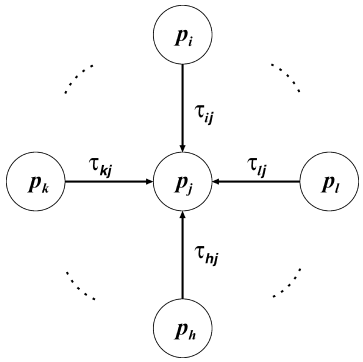

Fig. 6. Reputation ρ_j.

not obtain a satisfiable reply from the serving peer p_k, that is, the satisfiability σ_{ik} obtained from the serving peer p_k is 0.2. Here $Trust1(\tau_{ij}, \sigma_{ik}, \tau_{ik}, \alpha_i, \beta_i) = (1 - 1) \cdot 0.5 + 1 \cdot 0.5 \cdot \{0.9 \cdot 0.4 + (1 - 0.9) \cdot 0.2\} / 0.4 = 0.5 \cdot 0.38 / 0.4 = 0.475$. Thus, the trustworthiness τ_{ik} to the informing peer p_j is decreased to 0.475 from 0.5 since the peer p_j introduces the less trustworthy peer p_k to the peer p_i.

Each peer p_i is characterized by a tuple of the parameters $\langle \delta_i, \alpha_i, \beta_i \rangle$. Let p_i and p_j be a pair of peers. If $\delta_i < \delta_j$, a peer p_i is referred to as more *cumbersome* than another peer p_j. Here p_i does not like to ask another peer such as p_j. If $\alpha_i < \alpha_j$, a peer p_i is referred to as more *faithful* than another peer p_j. Here the peer p_i would like to trust the peer p_k even if the peer p_k returns an unsatisfiable reply if the peer p_k has so far been trustworthy. If $\beta_i < \beta_j$, a peer p_j is referred to as *closer* than p_i.

3.3 Ranking Factors

The *reputation* concept [Kamvar et al. 2003; Cuenca-Acuna et al. 2002] of each peer p_j shows how much a peer p_i is trusted by other peers in a peer-to-peer (P2P) overlay network. Let p_i be a requesting peer and p_j be its acquaintance peer. The reputation of the peer p_j shows how much the peer p_j is trusted by not only acquaintance peers of the peer p_i but also other peers that are not an acquaintance peer of the peer p_i (Figure 6). The reputation of a peer p_i might be influenced by malicious peers that give malicious trustworthiness to the peer p_j. In this article, in order to exclude the malicious trustworthiness, each peer p_i trusts only its acquaintance peer p_j because the peer p_i can directly communicate with the acquaintance peer p_j and can recognize how much each acquaintance peer can be trusted by the peer p_i itself. We discuss how much a requesting peer p_i perceives that the acquaintance peers trust an acquaintance peer p_j. Each peer p_i only takes into account how much its trustworthy acquaintance peer trusts the acquaintance peer p_j. Less trustworthy acquaintance peers and acquaintance peers of the peer p_j that are not an acquaintance of the requesting peer p_i are not considered. We introduce the *ranking factor* $\rho_{ij}(\langle o, \Box, op \rangle)$ of a requesting peer p_i to an acquaintance peer p_j, which shows how much an acquaintance peer p_j of a requesting peer p_i is

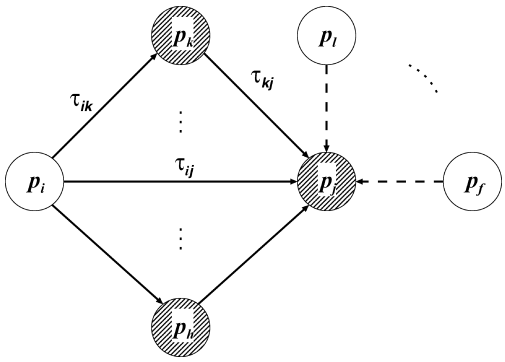

Fig. 7. Ranking factor ρ_{ij}.

trusted with respect to an access request $\langle o, \Box, op\rangle$. In this article, the ranking factor $\rho_{ij}(\langle o, \Box, op\rangle)$ depends on how much a trustworthy acquaintance peer p_k of a requesting peer p_i trusts an acquaintance peer p_j of the requesting peer p_i, that is, $\tau_{ik}(\langle o, \Box, op\rangle) \cdot \tau_{kj}(\langle o, \Box, op\rangle)$ with respect to an access request $\langle o, \Box, op\rangle$ [Watanabe et al. 2006].

Let τ_{ij} and σ_{ij} stand for the trustworthiness $\tau_{ij}(\langle o,\Box,op\rangle)$ and the satisfiability $\sigma_{ij}(\langle o,\Box,op\rangle)$ of a requesting peer p_i to an acquaintance peer p_j with respect to an access request $\langle o,\Box,op\rangle$, respectively, for simplicity. Each peer p_k distributes the trustworthiness τ_{kj} for each acquaintance peer p_j to every acquaintance peer in the view $view(p_k)$ (Figure 7) as discussed later. Each peer p_i calculates the ranking factor ρ_{ij} to an acquaintance peer p_j from the trustworthiness τ_{kj} received from each acquaintance peer p_k and the trustworthiness τ_{ik} for the peer p_k calculated in p_i by using Equation (3).

$$Rank0(p_i, p_j, \langle o, \Box, op\rangle) := \frac{\sum_{p_k \in view(p_i) \wedge \tau_{ik} \geq \epsilon_i} \sqrt{\tau_{ik}(\langle o, \Box, op\rangle) \cdot \tau_{kj}(\langle o, \Box, op\rangle)}}{|\{p_k \in view(p_i) \mid \tau_{ik}(\langle o, \Box, op\rangle) \geq \epsilon_i\}|}. \quad (3)$$

In the calculation of the ranking factor in Equation (3), only the trusted acquaintance peer p_k is considered where $\tau_{ik} \geq \epsilon_i$ for some constant ϵ_i ($0 \leq \epsilon_i \leq 1$). This means that the requesting peer p_i perceives that p_i can trust an acquaintance peer p_k if $\tau_{ik} \geq \epsilon_i$. The trustworthiness τ_{kj} of a less trustworthy acquaintance p_k to the peer p_j is removed in the calculation of the ranking factor ρ_{ij}. In addition, the trustworthiness τ_{kj} of an acquaintance peer p_k to the peer p_j is projected to the trustworthiness τ_{ik} of the requesting peer p_i to the acquaintance peer p_k. If an acquaintance peer p_k is more trustworthy to the peer p_i, the requesting peer p_i has more trust in what the acquaintance peer p_k mentions about the peer p_j. The ranking factor $\rho_{ij}(\langle o, \Box, op\rangle)$ is changed with the function $Rank0(p_i, p_j, \langle o, \Box, op\rangle)$ after updating the trustworthiness information in the view $view(p_i)$.

Let us consider an example where 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 $view(p_1) = \{p_0, p_2, p_3, p_4, p_5\}$. Suppose the trustworthiness τ_{0j} for each peer p_j is given as $\tau_{01}(\langle o, \models, op\rangle) = 0.7$, $\tau_{11}(\langle o, \models, op\rangle) = 1.0$, $\tau_{02}(\langle o, \models, op\rangle) = 0.7$, $\tau_{03}(\langle o, \models, op\rangle) = 0.0$, $\tau_{04}(\langle o, \models, op\rangle) = 0.4$, $\tau_{21}(\langle o, \models, op\rangle) = 0.8$, $\tau_{31}(\langle o, \models, op\rangle) = 0.9$, $\tau_{41}(\langle o, \models, op\rangle) = 0.6$, and

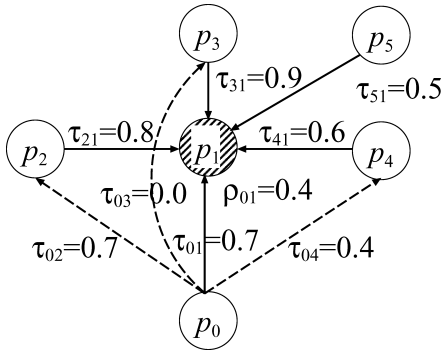

Fig. 8. Example 1 of the ranking factor ρ_{01}.

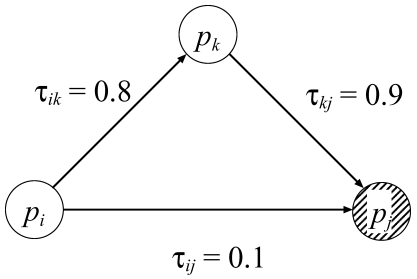

Fig. 9. Example 2 of the ranking factor ρ_{ij}.

$\tau_{51}(\langle o, \models, op\rangle) = 0.5$ as shown in Figure 8. Let ϵ_i be 0.1. Here the ranking factor $\rho_{01}(\langle o, \models, op\rangle)$ of the peer p_0 to the acquaintance peer p_1 is $(\sqrt{0.7 \cdot 1.0} + \sqrt{0.7 \cdot 0.8} + \sqrt{0.4 \cdot 0.6})/3$=0.692. The trustworthiness $\tau_{51}(\langle o, \models, op\rangle)$ is not considered in the ranking factor ρ_{01} since the peer p_5 is not an acquaintance peer of the peer p_0. According to the traditional reputation concept, the ranking factor ρ_{01} is given as $(\tau_{21} + \tau_{31} + \tau_{41} + \tau_{51})/4 = (0.8 + 0.9 + 0.6 + 0.5)/4 = 0.7$. If the peer p_5 is not trustworthy for the peer p_0, for example, p_5 is malicious, the ranking factor ρ_{01} is not reliable. In addition, the acquaintance peer p_3 is not trusted by the peer p_0, that is, $\tau_{03} = 0.0$. The trustworthiness τ_{31} is not considered in the ranking factor ρ_{01} even if the peer p_3 trusts the peer p_1 but the peer p_0 does not trust the peer p_3. In the ranking factor, only the trustworthiness of a trustworthy acquaintance peer is considered.

Let us consider three peers p_i, p_j, and p_k that are acquaintances of each other as shown in Figure 9. Suppose the peer p_i has less trusts in the peer p_j, say with the trustworthiness $\tau_{ij} = 0.1$. On the other hand, the peer p_i trusts the acquaintance peer p_k and the peer p_k trusts the peer p_j, say, $\tau_{ik} = 0.8$ and $\tau_{kj} = 0.9$. Here the peer p_j is more trusted according to the opinion of the peer p_k, that is, $\tau_{ik} \cdot \tau_{kj} = 0.8 \cdot 0.9 = 0.72$, but is less trusted than its own opinion of the peer p_i, that is, $\tau_{ij} = 0.1$. In real life, each person finally makes a decision based on his/her opinion even if other people have different opinions. A peer p_i first removes acquaintances' opinions quite different from its own opinion to an acquaintance peer p_j. Watanabe et al. [2006] discussed the ranking factor

with the deviation d based on this rule. We introduce the following equation to obtain the ranking factor $\rho_{ij}(\langle o, \square, op\rangle)$.

$$T_{ikj} = \begin{cases} \sqrt{\tau_{ik}(\langle o, \square, op\rangle) \cdot \tau_{kj}(\langle o, \square, op\rangle)} \\ \quad if \sqrt{\mid \tau_{ij}(\langle o, \square, op\rangle)^2 - \tau_{ik}(\langle o, \square, op\rangle) \cdot \tau_{kj}(\langle o, \square, op\rangle) \mid} \leq \varphi_i. \\ 0 \qquad\qquad otherwise. \end{cases} \tag{4}$$

$$Rank1(p_i, p_j, \langle o, \square, op\rangle) := \frac{\sum_{p_k \in view(p_i)} T_{ikj}(\langle o, \square, op\rangle)}{\mid \{p_k \in view(p_i) \mid T_{ikj} \neq 0\} \mid}. \tag{5}$$

Here φ_i is a constant ($0 \leq \varphi_i \leq 1$). In Figure 8, let τ_{01} be 0.7. $T_{021} = \sqrt{\tau_{02} \cdot \tau_{21}} = \sqrt{0.7 \cdot 0.8} = 0.748$ and $T_{041} = \sqrt{\tau_{04} \cdot \tau_{41}} = \sqrt{0.4 \cdot 0.6} = 0.490$. Let φ_0 be 0.5. $\sqrt{\mid \tau_{02} \cdot \tau_{21} - \tau_{01}^2 \mid} = \sqrt{\mid 0.56 - 0.49 \mid} = \sqrt{0.07} = 0.265 \leq 0.5$. $\sqrt{\mid \tau_{04} \cdot \tau_{41} - \tau_{01}^2 \mid} = \sqrt{\mid 0.24 - 0.49 \mid} = \sqrt{0.25} = 0.5 \leq 0.5$. The ranking factor ρ_{01} is $Rank1(p_0, p_1, \langle o, \square, op\rangle) = (\sqrt{0.8 \cdot 0.7} + \sqrt{0.6 \cdot 0.4})/2 = 0.619$. If $\varphi_0 = 0.3$, $Rank1(p_0, p_1, \langle o, \square, op\rangle) = \sqrt{0.8 \cdot 0.7} = 0.748$. Thus, only the acquaintance peer p_j where trustworthiness τ_{jk} is closer to the requesting peer p_i is taken into account if φ_0 is getting smaller.

A function $Rank(p_i, p_j, \langle o, \square, op\rangle)$ means either $Rank0(p_i, p_j, \langle o, \square, op\rangle)$ or $Rank1\ (p_i, p_j, \langle o, \square, op\rangle)$. A peer p_i takes the function $Rank1$ if the peer p_i is a larger number self-confidence and has a large number of acquaintance peers.

4. IMPLEMENTATION

We discuss how to maintain the trustworthiness and ranking factor in each peer.

4.1 Inter-peer Communication

A peer communicates with acquaintance peers by exchanging request and reply messages in an overlay network. Suppose a peer p_i sends an access request $\langle o, \square, op\rangle$ to an acquaintance peer p_j for an object o, a method op, and a P2O relation $\square$. A request message q is composed of the following fields.

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

In this article, we assume there is some mechanism to assign a unique identifier to each message, that is, $m_1.id \neq m_2.id$ for every pair of different messages m_1 and m_2. Each time a message m passes a peer, $m.TTL$ is decremented by one. If $m.TTL = 0$, a message m is discarded.

Suppose that a peer p_i receives a request message q for an access request $\langle o, \square, op\rangle$ from an acquaintance peer p_j. The peer p_i checks if p_i locally supports

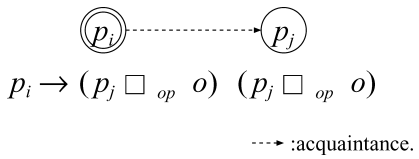

Fig. 10. Access.

service required by the access request $\langle o, \square, op \rangle$. For example, the peer p_i first looks for a target object o $(= q.oid)$ in the local database for a detection request $\langle o, |, \_ \rangle$. The peer p_i sends a *reply* message r for the request q to the requesting peer p_j if the peer p_i is an object holder peer, that is, $p_i \mid o$. Otherwise, the requested peer p_i forwards the access request $r(\langle o, \square, op \rangle)$ to the acquaintance peers of p_i.

A *reply* message r of a request message q includes the following fields.

—$r.id$ = identifier of the reply message r;
—$r.src$ = source peer which sends the reply message r;
—$r.qid$ = identifier $q.id$ of the access request q, that is, 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 $r.src$ to the target peer $r.sid$;
—$r.\tau$ = trustworthiness of $r.src$ to the target peer $r.sid$;
—$r.\rho$ = ranking factor of $r.src$ to the target peer $r.sid$.

4.2 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 acquaintance peers. A scheme of the acquaintance base AB_i is given a tuple $\langle pid, sid, oid, op, req, \sigma, \tau, \rho, \{iid\}, c \rangle$ of attributes. For a tuple t and attribute a in the acquaintance base AB_i, let $t.a$ denote a value of an attribute a in a tuple t. Here an attribute *pid* shows an identifier of an acquaintance peer of the peer p_i. An attribute *oid* indicates an identifier of a target object; *req* is a type $\square$ of access request $\in \{|, \vdash, \models\}$. An attribute *sid* is an identifier of a peer which supports service satisfying the request *req* on the object *oid*, that is, $\langle sid, q, oid \rangle$ *op* is a method. σ, τ, and ρ are the satisfiability, trustworthiness, and ranking factor of the peer p_i to the acquaintance peer *pid*, respectively. An attribute $\{iid\}$ shows a set of informing peers which informs the peer p_i of the acquaintance information Attributes $\langle p_j, \square_{op}, o \rangle$ where $sid = p_j$ and $req = \square_{op}$. Last, an attribute c is a counter showing how many times the tuple is accessed.

Suppose a peer p_i newly obtains an acquaintance (Figure 10) peer p_j which is a target peer of an access request $\langle o, \square, op \rangle$, that is, $p_i \rightarrow (p_j \ \square_{op} \ o)$. A tuple $\langle p_j, p_j, o, op, \square, \sigma_{ij}, \tau_{ij}, \rho_{ij}, \_, 0 \rangle$ is stored in the acquaintance base AB_i if no tuple $\langle p_i, p_j, o, op, \square, \cdots \rangle$ is in the acquaintance base AB_i. If a tuple $t = \langle p_i,$

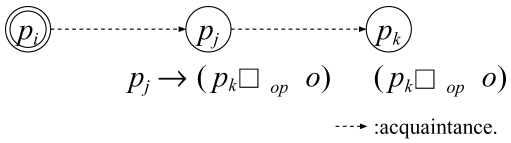

Fig. 11. Access.

AB_i	*pid*	*sid*	*oid*	*op*	*req*	σ	τ	ρ	{*iid*}	*c*
	p_j	p_k	*o*	*op*	□	σ_{ij}	τ_{ij}	ρ_{ij}	{p_k...}	c_j
	p_k	p_l	*o*	*op*	□	σ_{ik}	τ_{ik}	ρ_{ik}	{...}	c_k

Fig. 12. Acquaintance base AB_i.

$p_j, o, op, \square, \cdots\rangle$ is in the acquaintance base AB_i, the tuple t is updated as $t.\sigma = \sigma_{ij}, t.\tau = Trust0(t.\tau, \sigma_{ij}, \alpha_i)$, and $t.\rho = Rank(p_i, p_j, \langle o, \square, op\rangle)$. Here $\sigma_{ij} = \delta_i$ and $\tau_{ij} = Trust0(0, \sigma_{ij}, \alpha_i) = (1 - \alpha_i) \cdot \sigma_{ij}$. The ranking factor ρ_{ij} is obtained by $Rank(p_i, p_j, \langle o,\square,op\rangle)$.

Next, suppose a peer p_j is an acquaintance peer of a serving peer p_k where $p_k \square_{op} o$ and sends acquaintance information $p_j \rightarrow (p_k \square_{op} o)$ with the satisfiability σ_{jk}, trustworthiness τ_{jk}, and ranking factor ρ_{jk} to a peer p_i (Figure 11). On receipt of the acquaintance information from the peer p_k, a tuple $t = \langle p_j, p_k, o, op, \square, \dots \rangle$ is looked up in the acquaintance base AB_i. If not found, a tuple $\langle p_j, p_k, o, op, \square, \sigma_{ij}, \tau_{ij}, \rho_{ij}, \_, 0\rangle$ is stored in the acquaintance base 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 $\rho_{ij} = Rank(p_i, p_j, \langle o,\square,op\rangle)$. If found, the tuple t is updated as $t.\sigma = \delta_i\cdot\sigma_{jk}$, $t.\tau = Trust0(t.\tau, t.\sigma, \alpha_i)$, and $t.\rho = Rank(p_i, p_j, \langle o, \square, op\rangle)$. In addition, a tuple $\langle p_k, p_k, o, op, \square, \sigma_{ik}, \tau_{ik}, \rho_{ik}, \{p_j\}, 0\rangle$ is stored in the acquaintance base AB_i if $\langle p_j, p_k, o, op, \square, \dots \rangle$ is not in the acquaintance base AB_i. A tuple $\langle p_k, p_k, o, op, \square, \dots, \{p_j, \dots\}\rangle$ showing acquaintance information is associated with an informing peer p_j (Figure 12). If a tuple $t = \langle p_k, p_k, o, op, \square, \dots\rangle$ is in the acquaintance base AB_i, the tuple t is updated as $t.\sigma = \delta_i\cdot\sigma_{jk}$, $t.\tau = Trust0(t.\tau, t.\sigma, \alpha_i)$, $t.\rho = Rank(p_i, p_k, \langle o, \square, op\rangle)$, and $t.iid = t.iid \cup \{p_i\}$.

Suppose a tuple $t = \langle p_j, p_k, o, op, \square, \sigma, \tau, \rho, iid, c\rangle$ is updated in the acquaintance base AB_i, that is, $t.\sigma$ is changed. For each informing peer p_k in $t.iid$, a tuple $u = \langle p_k, p_l, o, op, \square, \dots\rangle$ is also changed as $u.\tau = Trust1(u.\tau, t.\sigma, \beta_i)$. As discussed before, the more satisfiable the peer p_i is for the informing peer p_k, the more trustworthy peer p_k is.

Each peer p_i first searches the acquaintance base AB_i for a tuple $t = \langle p_j, p_k, o, op, \square, \dots\rangle$. If found, $t.c$ is incremented by one. If $t.c$ gets larger than some

constant, $t.iid = \phi$ and $t.c = 0$. The peer p_i is informed of the acquaintance information $p_j \rightarrow (p_k \ \square_{op} \ o)$ by an informing peer p_j in $t.iid$. If the peer p_i communicates with the acquaintance peer p_k more times than some certain number, the peer p_i perceives the peer p_k to be its acquaintance peer and forgets about the informing peer p_k, that is, $t.iid = \phi$. That is, the trustworthiness of the informing peer p_j is not changed if the satisfiability to the peer p_k is obtained.

If the trustworthiness $t.\tau$ in the tuple t is updated, the trustworthiness in the tuple of the informing peer p_k ($\in t.iid$) 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 $\rho_{ik} = Rank(p_i, p_j, \langle o,\square,op \rangle)$. Suppose the peer p_i issues an access request $\langle o, \square, op \rangle$ to the peer p_k by using the acquaintance information tuple $t = \langle p_k, p_k, o_h, op, \square, \sigma_{ik}, \tau_{ik}, \rho_{ik}, \{p_j\} \rangle$ in the acquaintance base AB_i. Then the requesting peer p_i receives the reply from the peer p_k and obtains the satisfiability σ. Here the tuple is updated as $t.\sigma = \sigma$ and $t.\tau$ is changed with $Trust0(\tau_{ik}, \sigma, \alpha_i) = (1 - \alpha_i)\tau_{ik} + \alpha_i \cdot \sigma$. The ranking factor $t.\rho$ is changed with $Rank(p_i, p_j, \langle o,\square,op \rangle)$. In addition, the trustworthiness $t.\tau$ of a tuple $t = \langle p_j, p_k, o, op, \square, \sigma_{ij}, \tau_{ij}, \rho_{ij}, \{p_l\} \rangle$ in the acquaintance base AB_i is changed with $Trust1(\tau_{ij}, \sigma_{ik}, \beta_i) = (1 - \beta) \cdot \tau_{ij} + \beta \cdot \tau_{ij} \cdot Trust0(\tau_{ik}, \sigma, \alpha)/\tau_{ik}$. If $t.iid \neq \phi$, the trustworthiness $u.\tau$ of $u = \langle p_l, \ldots, \tau_{il}, \ldots \rangle$ in the acquaintance base AB_i is also changed for every peer p_l in $t.iid$ since the peer p_j is introduced to the requesting peer p_i by the informing peer p_l as discussed.

Since the peer p_i is a target peer of the object o, the peer p_i sends a reply message r such that $r.oid = o$, $r.sid = p_i$, and $r.\sigma = \sigma_{ii} = 1$, to the requesting peer p_j. If the peer p_i is not a target peer of the object o, that is, $p_i \not\square_{op} o$, the peer p_i searches the acquaintance base AB_i for tuples of the access request $\langle o,\square,op \rangle$. Suppose a tuple $\langle p_j, p_k, o, op, \square, \sigma_{ij}, \tau_{ij}, \rho_{ij}, p_f \rangle$ is found in the acquaintance base AB_i. Here $j = k$ if $p_i \rightarrow (p_k \ \square_{op} \ o)$. 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 the acquaintance base AB_i, the peer p_i decrements $q.TTL$ of a request message q by one. If $q.TTL \geq 1$, the peer p_i forwards the access request q to every acquaintance peer p_k except for the requesting peer p_j. The peer p_i waits for replies from the acquaintance peers. If $q.TTL = 0$, the peer p_i discards the request message q.

On receipt of a reply message r of the request q from an acquaintance peer p_j, a peer p_i updates the acquaintance base AB_i as follows.

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

The peer p_i waits for a reply message from every acquaintance peer to which the peer p_i sends a request message q. If the peer p_i receives every reply message or the timer expires, the peer 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 message q to the requesting peer p_j.

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

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 the acquaintance base AB_i. If AB_i is full, the tuple a cannot be added to the acquaintance base AB_i. A tuple b in the acquaintance base AB_i is selected and removed to make space to store the tuple a by the following rule:

[Selection rule]

(1) Select a tuple b where $b.\tau$ is the smallest in the acquaintance base 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 peer p_j, a requesting peer p_i obtains acquaintance information:

if $p_i \rightarrow (p_j \Box_{op} o)$, {
  $\sigma_{ij} = r.\sigma \cdot \delta_{ij}$;
  stAB$(p_i, \langle p_j, p_k, o, op, \Box, \sigma_{ij} \cdot \delta_i, Trust0(0, \sigma_{ij}, \alpha_i), 0, \_, 0 \rangle)$;
}
if $p_i \rightarrow (p_j \rightarrow (p_k \Box_{op} o))$, {
  $\sigma_{ij} = r.\sigma \cdot \delta_i$;
  stAB$(p_i, \langle p_j, p_k, o, op, \Box, \sigma_{ij}, Trust0(0, \sigma_{ij}, \alpha_i), 0, p_j, 0 \rangle)$;

if p_i is not *careful*, {
$\sigma_{ik} = \sigma_{ij}$;
stAB(p_i, $\langle p_k, p_k, o, op, \square, \sigma_{ik}, Trust0(0, \sigma_{ik}, \alpha_i), 0, p_j, 0\rangle$);
}
}

stAB(p_i**,** $\langle p_j, p_k, o, op, \square, \sigma, \tau, \rho, p_f, c\rangle$**)** {

if $((t =$ **findAB**$(p_j, o, op, \square)) \neq$ NULL), {
$\sigma_{ij} = t.\sigma \cdot \delta_i$;
upAB$(p_i, t, \sigma_{ij}, Trust0(t.\tau, \sigma_{ij}, \alpha_i), t.\rho, o, op, \square, t.iid \cup \{p_f\})$;

} **else** {
if AB_i is **full**, {
one tuple is selected and removed;
$\langle p_j, p_k, o, op, \square, o, \tau, \rho, \{p_f\}, 0\rangle$ is stored is AB_i;
return ;
}
}

if $t.iid = \phi$, **return**;
for every p_k **in** $t.iid$, {
$u =$ **findAB**$(p_k, o, op, \square)$;
if $u \neq$ NULL, {
$\tau_{ik} = Trust1(u.\tau, \sigma_{ij}, \beta_i)$;
$\rho_{ik} = Round(p_i, p_k, \langle o, \square, op\rangle)$;
upAB$(p_k, t, u.\sigma, \tau_{ik}, \rho_{ik}, o, op, \square, u.iid)$;
}

}
}

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

findAB (p_i**,** o**,** op**,** $\square$**)** {
if $t = \langle p_i, p_j, o, op, \square, \cdots, c\rangle$ 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 [Urban et al. 2001]. A peer-to-peer overlay network includes n (≥ 1) peers p_1,

$\ldots$, p_n. Initially, each peer p_i is in an acquaintance relation with l_i ($\leq n$) peers that are randomly selected. There are m (≥ 1) objects $o_1,\ldots,o_m$. Let $\mathbf{P}$ be a set $p_1,\ldots,p_n$ of the peers and $\mathbf{O}$ be a set $o_1,\ldots,o_n$ of the objects in a P2P overlay network. 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 in the peer set $\mathbf{P}$ as a requesting peer and an object o_h is also randomly selected in the object set $\mathbf{O}$ 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 object holder peers of the target 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 of peers are changed because the peers obtain new acquaintance information as discussed. Hence, acquaintance information is distributed to the more peers as more rounds are completed. However, since the volume of the acquaintance base AB_i of each peer p_i is limited, some acquaintance information might be lost due to the tuple replacement. Some acquaintance peer may hold inconsistent acquaintance information. A sequence of rounds is referred to as one *run*. In this evaluation, 100 runs in total are performed. We obtain the average values of the hit ratio and satisfiability for each round.

In the evaluation, we assume that there are 1,000 peers, that is, $n = 1{,}000$. Each peer p_i is initially related to three acquaintance peers, that is, $l_i = 3$. We assume each peer p_i can store at most five tuples in the acquaintance base AB_i, that is, $t_i = 5$. This means that each peer p_i can have at most five acquaintance peers. 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, that is, $m = 1$. The hit ratio for an access request is defined to be the probability that a target peer is detected. For the kth round, the number s (≤ 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. The satisfiability is obtained for each run. The average satisfiability of the kth round is calculated for 1000 runs. Figure 13 shows the hit ratio for $\zeta = 1$ [%] and $\zeta = 10$ [%]. The horizontal axis shows the number of runs. For the 10th round, the hit ratios are 0.4 and 0.96 for $\zeta = 1$ and 10[%], respectively.

Through interactions among peers, acquaintance information is propagated in the network. The more rounds there are, the higher the satisfiability must be. Figure 14 shows the satisfiability for $\zeta = 1$ [%] and 10 [%]. For $\zeta = 1$ [%], it takes about 10 rounds to propagate the target peer information to every peer, while it takes 25 rounds for $\zeta = 10$ [%].

As discussed before, tuples in the acquaintance base AB_i of each peer p_i are replaced with new tuples. There are five objects, $m = 5$. One object is taken

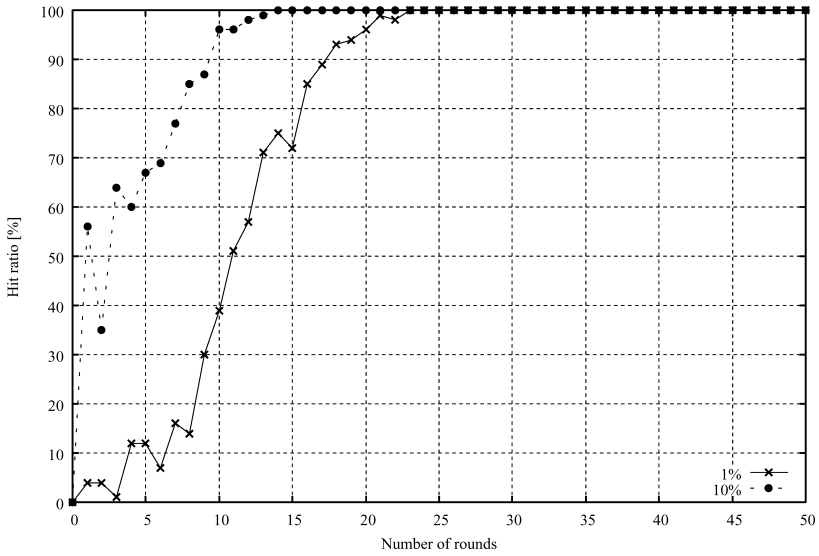

Fig. 13. Hit ratio.

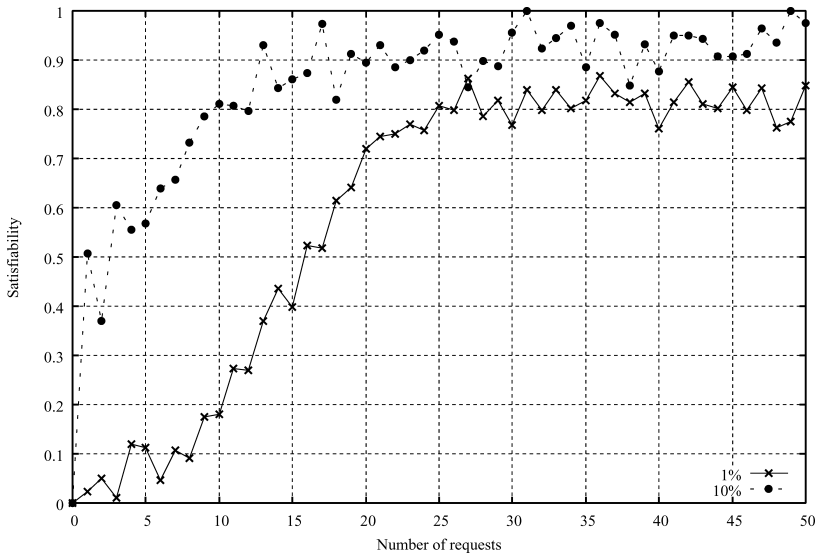

Fig. 14. Satisfiability.

and a requesting peer is randomly selected. For each object, 100 requests are issued, that is, a total of 500 requests are issued. Figures 15 and 16 show the hit ratio and satisfiability for sizes of the acquaintance base, that is, $t = 3$, 5, and 10 tuples. Through interaction with acquaintances, only more trustworthy peers are stored in each acquaintance base. Hence, the hit ratio is increased for the number of rounds. In Figure 15, the hit ratio is not always 1.0 because some acquaintance information is lost due to the replacement of tuples in the

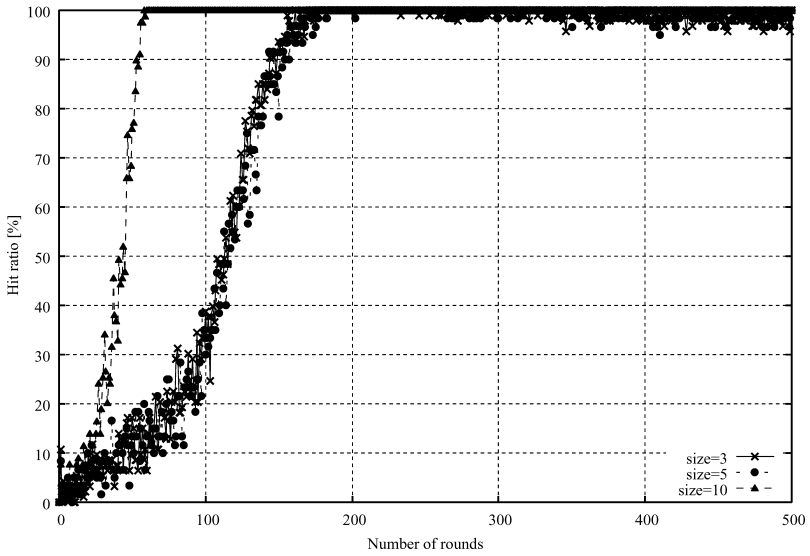

Fig. 15. Hit ratio for acquaintance base size.

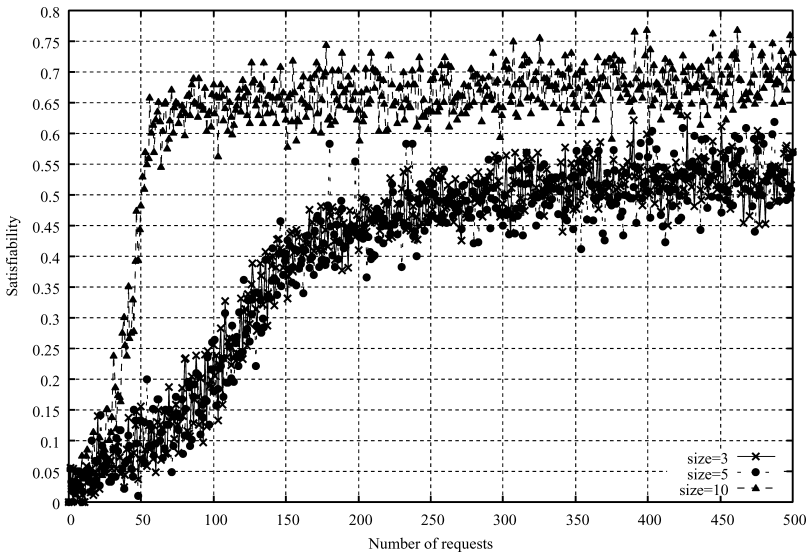

Fig. 16. Satisfiability for acquaintance base size.

acquaintance base. The larger the size of the acquaintance base, the higher the hit ratio and satisfiability.

6. CONCLUDING REMARKS

In fully distributed peer-to-peer overlay networks, each peer has to find a target peer and manipulate objects through communication with the acquaintance peers. It is critical to find trustworthy acquaintances since some acquaintace

peers may hold obsolete service information and may be faulty. We discussed how each peer trusts acquaintance peers in a peer-to-peer overlay network. First, types of acquaintance relations are defined with respect to the types of service of each peer, object holder peer, permission holder peer, and intermediated peer. In addition to finding where a target object exists, a requesting peer has to find an authorized acquaintance peer to obtain the access right and a manipulation peer that 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 peer and the ranking factor of each peer by aggregating the satisfiability obtained through each interaction with the acquaintance peer. We defined two types of ranking factors ρ_{ij} of a requesting peer p_i to an acquaintance peer p_j to show how much the acquaintance peer p_j is trusted by trustworthy acquaintance peers of the peer p_i. We discussed how each peer behaves to obtain the trustworthiness and the ranking factor of its acquaintance peers by issuing access requests to and receiving replies from the acquaintance peers. We evaluated how the hit ratio and satisfiability of acquaintance peers are changed through interactions among peers in the flooding algorithm.

REFERENCES

ANDROUTSELLIS-THEOTOKIS, S. AND SPINELLIS, D. 2004. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv. 36*, 4, 335–371.

CLARKE, I., SANDBERG, O., WILEY, B., AND HONG, T. W. 2000. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*. 46–66.

CRESPO, A. AND GARCIA-MOLINA, H. 2002. Routing indices for peer-to-peer systems. In *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*. 23–32.

CUENCA-ACUNA, F. M., MARTIN, R. P., AND NGUYEN, T. D. 2002. PlanetP: Using gossiping and random replication to support reliable peer-to-peer content search and retrieval. Tech. Rep. DCS-TR-494, Rutgers University.

EGEMEN, T., DEEPA, N., AND HANAN, S. 2005. An efficient nearest neighbor algorithm for P2P settings. In *Proceedings of the National Conference on Digital Government Research*. 21–28.

FERRAIOLO, F. D., KUHN, D. R., AND CHANDRAMOULI, R. 2003. *Role-Based Access Control*. Artech House Publishers.

KAMVAR, D. S., SCHLOSSER, T. M., AND GARCIA-MOLINA, H. 2003. The Eigentrust Algorithm for reputation management in P2P networks. In *Proceedings of the 12th IEEE International Conference on World Wide Web*. 640–651.

LAMPORT, L., SHOSTAK, R., AND PEASE, M. 1982. The Byzantine generals problem. In *ACM Trans. Program. Lang. Syst.* 382–401.

LIU, Y., ZHUANG, Z., LI, X., AND LIONEL, M. N. 2004. A distributed approach to solving overlay mismatching problem. In *Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS)*. 132–139.

NAKAJIMA, Y., WATANABE, K., HAYASHIBARA, N., ENOKIDO, T., TAKIZAWA, M., AND DEEN, S. M. 2006. Trustworthiness in peer-to-peer overlay networks. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*. 86–93

NAPSTER. http://www.napster.com.

OMG INC. 1997. *The Common Object Request Broker: Architecture and Specification*. QED Publishing Co.

ORACLE CORPORATION 1999. *Oracle8i Concepts* Vol. 1.

RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SCHENKER, S. 2001. A scalable content-addressable network. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. 161–172.

RIPEANU, M. 2001. Peer-to-peer architecture case study: gnutella network. In *Proceedings of International Conference on Peer-to-Peer Computing (P2P'01)*. 99–100.

ROWSTRON, A. AND DRUSCHEL, P. 2001. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*.

STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, F., AND BALAKRISHNAN, H. 2003. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. on Netw. (TON) 11,* 1, 17–32.

SYBASE SQL SERVER. Sybase sql server. http://www.sybase.com/.

URBAN, P., DEFAGO, X., AND SCHIPER, A. 2001. Neko: A single environment to simulate and prototype distributed algorithms. In *Proceedings of the 15th International Conference on Information Networking (ICOIN-15)*. 503–511.

WATANABE, K., ENOKIDO, T., AND TAKIZAWA, M. 2006. Trustworthiness of acquaintances in peer-to-peer overlay networks. *Inter. J. High Perform. Comput. Netw. (IJHPCN)*. To appear.

WATANABE, K., ENOKIDO, T., TAKIZAWA, M., AND KIM, K. 2005a. Charge-based flooding algorithm for detecting multimedia objects in peer-to-peer overlay networks. In *Proceedings of IEEE 19th Conference on Advanced Information Networking and Applications (AINA'05). 1*, 165–170.

WATANABE, K., HAYASHIBARA, N., ENOKIDO, T., AND TAKIZAWA, M. 2005b. CBF: Look-up protocol for distributed multimedia objects in peer-to-peer overlay networks. *J. Intercon. Netw. 6*, 3, 323–344.

WATANABE, K. AND TAKIZAWA, M. 2006. Service oriented cooperation among trustworthy peers. *J. Intercon. Netw. 1*, 4, 507–533.

ZHAO, B. Y., KUBIATOWICZ, J., AND JOSEPH, A. D. 2001. Tapestry: An infrastructure for fault-resilient wide-area location and routing. Tech. rep. UCB/CSD-01-1141, University of California, Berkeley.

Received March 2006; revised November 2006; accepted May 2007