IPSJ SIG Technical Report

Vol.2010-DPS-142 No.17
Vol.2010-CSEC-48 No.17
2010/3/4

Trustworthiness among Peer Processes in Distributed Agreement Protocol

AILIXIER AIKEBAIER,†1 VALBONA BAROLLI,†1
TOMOYA ENOKIDO†2 and MAKOTO TAKIZAWA †1

Nowadays more and more information systems are being shifted to distributed architectures because of the benefits like scalability, autonomy, and faulty-tolerance implied from the essence of the distributed systems. Here, every process is peer and cooperates with other peers to achieve common goal. In order to do that, peers have to efficiently and flexibly make an agreement on one common value which satisfies an agreement condition. In this paper, we consider a distributed group of multiple peers with no centralized coordination. We introduce a novel approach to efficiently making an agreement where each peer sends a package of multiple possible values to the other peers at each ongoing round. By exchanging multiple possible values at once, we can significantly reduce the total number of messages. The time and network resources are mostly spent in the value exchange phase. If we can reduce the time and number of messages to exchange values among peers, we can improve the efficiency of the agreement protocol. In order to efficiently exchange value packages among peers, we take advantage of the multipoint relaying mechanism to reduce the number of duplicate re-transmissions. Although we can significantly reduce the re-transmitted values, we have to realize the fault-tolerancey of the system. In addition to improving the reliability of the multipoint relaying mechanism, we newly introduce the trustworthiness among peers. By taking into account the trustworthiness of the peer, each peer broadcasts values through the trusted neighbors to the other peers. Here, the transmission fault which causes by untrusted, unreliable peers can be prevented.

1. Introduction

There are two typical models of information systems. One is the cloud computing model[10] where a huge number of server computers are virtualized to one system and are used by thin clients. The other model is the peer-to-peer (P2P) model [15),22)] where every computer is peer because it can be a server and a client. In this paper, we consider a fully distributed P2P system where there is no centralized coordinator and each peer process (peer) is autonomous and independent. In P2P applications like Intelligent Decision Advisor (IDA), Distributed Decision Making (DDM), Computer Supported Cooperative Work (CSCW), a group of multiple peers are requires to make an agreement on a common value, for example, to fix a date of meeting, best position of build a building and so on. There are many discussions on how to make an agreement on one value out of values shown by the peers in presence of types of faults [7),11),13),14)]. They do not discuss relations among values to be shown by each peer. The authors [2),3),20)] discuss types of precedent relations on values to show what value a peer can take after a value. In the agreement protocol, it is significant for each peer to decide on which value to show to the other peers if there are multiple possible values. The authors [4)] discuss the coordination strategies, forward, backward, mining, and observation strategies to efficiently make an agreement among peers. Some combinations of strategies taken by peers are inconsistent. We define what combinations of strategies are consistent, and discuss how the peers resolve the inconsistency of the strategies and take consistent strategies [4)].

In the agreement protocols[2),3),20)], each peer exchanges one value with the other peers at each round. Multiple rounds are spent by sending one value at each round in the traditional agreement protocols. In time critical applications, the final decision on a proposing opinion has to be made within some limited time period. Thus, it is significant to discuss how to reduce the overall time overhead of the agreement protocol. In addition, we have to reduce the number of messages sent by the peers at each round. Thus, it is also really important to consider how to reduce re-transmitted messages. Furthermore, we have to reduce the number of rounds to exchange values. In this paper, we discuss a novel *multi-value exchange* (*MVE*) *protocol* to effectively reduce the number of rounds which it takes to enrich the agreement condition. Here, where each peer p_i shows the other peers a package of multiple possible values at each round. Values in a package are ordered in the preference which is pre-decided according to the needs of the individual peer. Thus, each peer can collect values to be exchanged at not only current round but also upcoming rounds, each peer can find a tuple of values which satisfy the agreement condition in the family of the packages. We can reduce total time to exchange multiple values among peers. Furthermore, we can increase the probability that every peer makes an agreement.

On the other hand, to reduce the number of re-transmissions in during the message exchange among peers, we take advantage of the *multipoint relying* mechanism [17)] which

†1 Seikei University
†2 Rissho University

can significantly reduce the number of re-transmitted messages. However, we have to sacrifice some level of reliability of the system. In fact, the reliability of the value exchange will directly imply whether or not peers can finally make an agreement. Here, to improve the fault-tolerance of the multipoint relying mechanism, we newly introduce the trustworthiness among peers among peers. Each peer broadcasts values by sending them through the trusted neighbors to the other peers. The transmission fault caused by unreliable peers can be largely prevented.

In section 2, we discuss the multi-value exchange (MVE) scheme in the agreement protocol. In section 3, we briefly present the multipoint relay (MPR) mechanism. In section 4, we discuss trustworthiness of peers and improve MPR by taking advantage of the trustworthiness concept.

2. Multi-value Exchange (MVE) Scheme

Let us consider a group G of peers $p_1, \ldots, p_n$. The domain D_i is a set of possible values which a peer p_i can take. A peer p_i takes a value v_1. There are values which p_i can take. A value v_1 *existentially* (*E*-) *precedes* another value v_2 in a peer p_i ($v_1 \rightarrow_i^E v_2$) if and only if (*iff*) p_i is allowed to take v_1 after v_2[1)–3)]. We assume the precedent relation $\rightarrow_i^E$ is transitive. v_1 and v_2 are *E- incomparable* in p_i ($v_1 |_i^E v_2$) iff neither $v_1 \rightarrow_i^E v_2$ nor $v_2 \rightarrow_i^E v_1$. The *preferentially* (*P*-) *precedent* relation $\rightarrow_i^P$[1)–3)] is also defined. Let $Corn_i(x)$ be a set of values which a peer p_i can take after a value x, i.e. $\{y \mid x \rightarrow_i^E\}$.

Suppose each peer p_i can have a subset I_i of initial values ($I_i \subseteq D_i$) which p_i would like to take in the agreement procedure. Let PV_i be a set of values $\cup_{x \in I_i} Corn_i(x)$, which shows a subset of possible values which a peer p_i can take at the initial round. If there is a satisfiable tuple $\langle v_1, \ldots, v_n \rangle \in PV_1 \times \cdots \times PV_n$ for the agreement condition AC, every peer can make an agreement on the tuple. Here, the group G of the peers $p_1, \ldots, p_n$ are referred to as *agreeable* for the agreement condition AC. Otherwise, the peers cannot make an agreement for AC. Suppose there are a pair of satisfiable tuples $\langle x_1, \ldots, x_n \rangle$ and $\langle y_1, \ldots, y_n \rangle$ in the direct product $PV_1 \times \cdots \times PV_n$. If $x_i \rightarrow_i^E y_i$ or $x_i |_i^E y_i$ for every peer p_i, the tuple $\langle x_1, \ldots, x_n \rangle$ is referred to as *precedes* the tuple $\langle y_1, \ldots, y_n \rangle$. Suppose a pair of satisfiable tuples $\langle x_1, \ldots, x_n \rangle$ and $\langle y_1, \ldots, y_n \rangle$ are not preceded. If $x_i \rightarrow_i^P y_i$ or $x_i |_i^P y_i$ for every p_i, the tuple $\langle x_1, \ldots, x_n \rangle$ is more *preferable* than the tuple $\langle y_1, \ldots, y_n \rangle$.

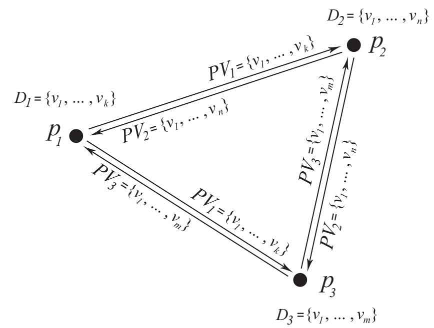

Fig. 1 Maximal-value exchange.

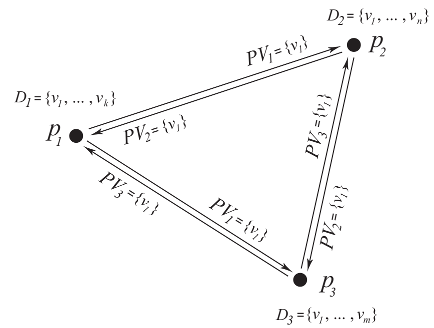

Fig. 2 Single-value exchange.

In the basic protocol, each peer p_i exchanges the value set PV_i with the other peers. Then, each peer p_i finds the most preceded, preferable tuple in the direct product $PV_1 \times \cdots \times PV_n$. It takes just one round to make an agreement. This is referred to as *maximal value exchange* (XVE) scheme [Figure 1]. At the other extreme, each peer sends only one value in PV_i like the simple protocols[1)–3),20)]. Each peer p_i has to show a value x after y where $y \rightarrow_i^E x$. This is referred to as *single value exchange* (SVE) scheme [Figure 2]. It takes each peer more than one round to show multiple possible values to the other peers. Furthermore, depending on an order in which each peer shows values to the other peers, the peers may not make an agreement even if the peers are agreeable. For example, a peer p_1 has a pair of possible values a and b and another peer

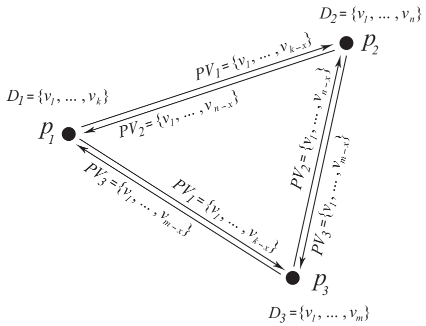

Fig. 3 Multi-value exchange.

p_2 has a pair of possible values b and c. If p_1 and p_2 show values b and c, respectively, the peers show different values a and c. Here, the peers p_1 and p_2 cannot make an agreement even if the peers have the satisfiable value b.

There is a *multi-value exchange* (MVE) [Figure 3] scheme in between the extreme cases XVE and SVE. Here, each peer p_i sends a subset of PV_i to the other peers. The more number of values are exchanged at each round, the shorter it takes to make an agreement and the higher possibility every peer makes an agreement. On the other hand, the more communication overhead and processing overhead might be implied. There is a trade off point between size of a package and the time spending on exchange packages.

In order to more efficiently make an agreement among peers, we discuss the *multi-value exchange* (MVE) scheme. At each round t, each peer p_i sends a *package* V_i of possible values to the other peers. In the package, values are ordered in the preference. The top value of the package is the most preferable value named primary one. The others are secondary ones. On receipt of the package V_j from every peer p_j, each peer p_i finds a satisfiable tuple of values in a collection of the packages $V_1, \ldots, V_n$. For example, suppose there are a pair of peers p_1 and p_2. The peer p_1 sends a package $V_1 = \{a, b\}$ and p_2 sends $V_2 = \{b, c\}$. On receipt of the package V_2 from p_2, the peer p_1 finds that the other peer p_2 can also take the value b. Then, the peers p_1 and p_2 agree on the value b in the all agreement condition. Thus, by taking advantage of the MVE scheme, each peer p_i obtains one or more than one possible value from every other peer at one round. Then, each peer p_i can find a satisfiable tuple of values in a collection of the packages $V_1, \ldots, V_n$ which p_i has received from the other peers if the peers are agreeable. Thus, we can significantly reduce the overall time overhead of the agreement protocol and increase the possibility that a group of agreeable peers make an agreement.

In this paper, we consider a static group where each peer p_i does not change the domain D_i and the precedent relations $\rightarrow_i^E$ and $\rightarrow_i^P$. Here, each peer p_i can collect a set V_i of possible values which p_i can take, $V_i \leq D_i$.

In the XVE scheme, each peer sends the whole set V_i to the other peers at one round. Then, each peer p_i tries to find a satisfiable tuple of values in the family of the sets $V_1, \ldots, V_n$. On the other hand, each peer p_i cannot send the set V_i at one round, like in the MVE scheme. For a pair of subsets V_{ij} and V_{ik}, V_{ij} *E-precedes* V_{ik} ($V_{ij} \rightarrow_i^E V_{ik}$) iff $v_1 \rightarrow_i^E v_2$ or $v_1 \mid_i^E v_2$ for every pair of values v_1 in V_{ij} and v_2 in V_{ik}. $V_{ij} \mid_i^E V_{ik}$ if neither $V_{ij} \rightarrow_i^E V_{ik}$ nor $V_{ik} \rightarrow_i^E V_{ij}$. Thus, a collection of the subsets $V_{i1}, \ldots, V_{il_i}$ are partially ordered in the E-precedent relation $\rightarrow_i^E$. As discussed in the SVE scheme, the peer p_i has to show a subset so that the E-precedent relation is satisfied.

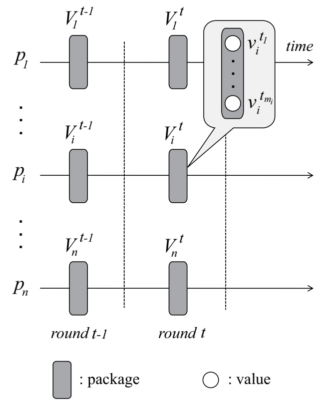

Fig. 4 Multi-value exchange.

The peer p_i has to send a subset of the set V_i at each round. Thus, the set V_i has to be decomposed into subsets $V_{i1}, \ldots, V_{il_i}$ ($l_i > 1$). At each round t, a peer p_i receives packages $V_1^t, \ldots, V_n^t$ from the peers $p_1, \ldots, p_n$, respectively, as shown in Figure 4. Here, if there is a tuple $\langle v_1, \ldots, v_n \rangle \in V_1^t \times \cdots \times V_n^t$ of values which satisfy the agreement condition AC, every peer p_i makes an agreement on the tuple $\langle v_1, \ldots, v_n\rangle$ and then takes an agreement value from the tuple. For example, the values in the tuple are the same, $v_1 = \ldots = v_n = v$ and the value v is an agreement value in the *all* agreement condition. In the *majority* condition, a majority value in the tuple is taken as an agreement value.

There may be multiple tuples in $V_1^t \times \cdots \times V_n^t$ which satisfy the agreement condition AC. Here, let $ord(v_j)$ denote the P-preferent order of a value v_j in a package V_j^t, i.e. $ord(v_j) > ord(v_j^{'})$ if $v_j \rightarrow_i^P v_j^{'}$, i.e. p_j prefers v_j to $v_j^{'}$. For example, $ord_i(v_j^{tk})$ is k in a package $V_i^t = \langle v_j^{t1}, \ldots, v_j^{tm_j}\rangle$. Let $\langle x_1, \ldots, x_n\rangle$ and $\langle y_1, \ldots, y_n\rangle$ be a pair of tuples in the direct product $V_1^t \times \cdots \times V_n^t$ of the packages. Here, a tuple $\langle x_1, \ldots, x_n\rangle$ is more preferable than another tuple $\langle y_1, \ldots, y_n\rangle$ in a peer p_i if $\sum_{k=1}^n ord_i(x_k) < \sum_{j=1}^n ord_i(y_j)$. If there is no tuple which is more preferable to a tuple $\langle x_1, \ldots, x_n\rangle$, the tuple $\langle x_1, \ldots, x_n\rangle$ is referred to as *maximally* preferable. If there are multiple maximally preferable tuples which satisfy the agreement condition AC, each peer p_i takes one of the maximally preferable tuples. For example, a tuple whose i the element is the most preferable in a peer p_i whose identity is the smallest is taken.

If there is no tuple satisfying the agreement condition AC, each peer p_i finds values which is E-preceded by the primary value v_i^{t1} in the package V_i^t. At round $t+1$, each peer p_i sends a package V_i^{t+1} where every value is E-preceded by the primary value v_i^{t1} in V_i^t. In this paper, we assume each package V_i^t can include at most some number K ($\geq$1) of the possible values; the primary value v_i^{t1} and secondary values $v_i^{t2}, \ldots, v_i^{tK}$ in order to increase the performance and make the implementation simple.

The application layer of each individual peer makes a decision on what value the peer can take at the next round. In addition, the agreement condition AC of the group is decided according to the purpose of the group like majority decision and so on.

Suppose a peer p_1 takes a value a at round t and can take values b, c, and d at round $t+1$, i.e. $a \rightarrow_i^E b, c, d$. Suppose another peer p_2 takes a value e at round t and can take values d and c at round $t+1$. In the traditional protocols, if a pair of the peers p_1 and p_2 take the value d at the same round, the processes p_1 and p_2 can agree on the value d. Suppose the peer p_1 takes the value d but the peer p_2 takes the value c at round $t+1$, respectively. Then, the peers p_1 and p_2 take the values c and d, respectively. Here, the peers p_1 and p_2 cannot make an agreement although both the processes p_1 and p_2 can take values c and d. In the multi-value exchange scheme, the peers p_1 and p_2 send the packages $V_1 = \langle b, c, d\rangle$ and $V_2 = \langle c, d\rangle$, respectively, to one another. Then, the peers p_1 and p_2 find a pair of satisfiable tuples $\langle c, c\rangle$ and $\langle d, d\rangle$ in the packages V_1 and V_2. Here, the value c is taken because the peers p_1 and p_2 prefer the value c to d, i.e. the tuple $\langle c, c\rangle$ is more preferable than $\langle d, d\rangle$.

3. Multipoint Relaying (MPR) Mechanism

In a group of multiple peers, each peer has to broadcast a message with a package of values to all the other peers. In one approach to broadcasting a message in a P2P overlay network, a peer first sends a message to the neighbor peers. On receipt of a message, a peer forwards the message to the neighbor peers. Thus, a message floods in the network. This is a pure flooding scheme[18)]. However, the pure flooding scheme implies the huge network overhead due to the message explosion.

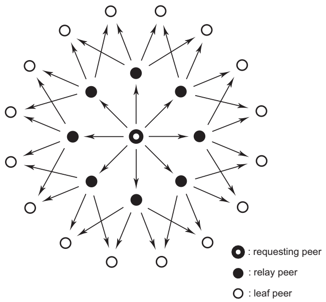

Fig. 5 Pure flooding.

The concept of "multipoint relaying (MPR)" is developed to reduce the number of *duplicate transmissions* while each peer forwards a message to the neighbor peer[17)]. Here, on receipt of a message, a peer forwards the message to all the neighbor peers but only some of the neighbor peers forward the message differently from the pure flooding scheme. By taking into consideration the second neighbor peers in addition to the first neighbor peers, each peer obtains a subset of the first neighbor peers which forward the message. The other neighbor peers which are not selected just receive the message and do not forward it. The number of messages transmitted can be significantly reduced. The MPR provides an adequate solution to reduce the overhead to broadcast messages.

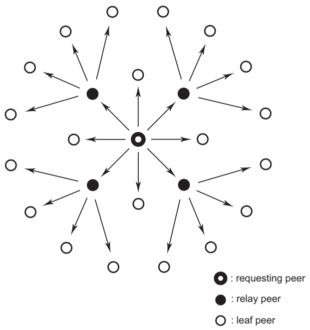

Fig. 6 Multipoint relays.

A neighbor peer p_j of a peer p_i, which forwards a message to its neighbor peer, is referred to as *relay peer* of the peer p_i. The other neighbor peers are *leaf* peers of p_i. Every leaf peer p_k just receives a message from p_i which every forward peer forwards the message to the neighbor peers. Let $N(p_i)$ be a set of one-hope neighbor peers of a peer p_i. A set of its the second neighbor peers of p_i is denoted by $N^2(p_i)$. $N^2(p_i)$ = $\cup_{p_j \in N(p_i)} N(p_j)$. Let $R(p_i)$ and $L(p_i)$ be collections of replay peers and leaf peers of a peer p_i, respectively. Here, $N(p_i)$ = $R(p_i) \cup L(p_i)$ and $R(p_i) \cap L(p_i)$ = ϕ. The following condition is required to hold:

- $N^2(p_i)$ = $\cup_{p_j \in R(p_i)} N(p_j)$

That is, a message sent by a peer p_i can be delivered to every second neighbor peer of p_i which only the relay peer of p_i forward the message to second neighbor peer of p_i. Here, we define the *coverage* of a peer p_i:

- A peer p_j is referred to as *covered* by a peer p_i iff $p_j \in N(p_i)$ or p_j is covered by some relay peer $p_k \in R(p_i)$.

A collection of peers covered by a peer p_i is referred to as subnetwork *covered* by p_i. The efficient algorithm for selecting multipoint relays[17] is proposed. Here, each peer p_i is assumed to know the second neighbor peers. Let $MPR(p_i)$ be a set of selected relay peers of a peer p_i. An algorithm for selecting $MPR(p_i)$ is shown as follows:

1. Start with an empty multipoint relay set $MPR(p_i)$, $MPR(p_i)$ = ϕ. S = $N^2(p_i)$. F = $N(p_i)$.
2. Select a neighbor peer p_j in $N(p_i)$ where $N(p_j) \cap N(p_k)$ = ϕ for every other first neighbor peer p_k in F and add the first neighbor peer p_j to the multipoint relay set $MPR(p_i)$. If found, $MPR(p_i)$ = $MPR(p_i) \cup \{p_j\}$, S = S - $N(p_j)$, and F = F - $\{p_j\}$, go to step 2 if F = ϕ.
3. While $S \neq \phi$, do the following steps:
   (a) For each peer p_j in F, compute the number $U(p_j)$ of peers which p_j covers in the set S, $U(p_j)$ = $N(p_j) \cap S$.
   (b) Add the peer p_j to $MPR(p_i)$ where $|U(p_j)|$ is the maximum, S = S - $U(p_j)$, F = F - $\{p_j\}$, $N(p_j)$ = $U(p_j)$.
4. For every peer p_j in F, $N(p_j)$ = ϕ, i.e. p_j is a leaf peer.

Hence, for each neighbor peer p_j in $N(p_i)$, $N(p_i)$ shows the neighbor peer of p_j. If p_j is a leaf peer, $N(p_i)$ = ϕ. For each neighbor peers p_j in $N(p_i)$, the algorithm is applied to obtain a set $MPR(p_j)$ of relay peers of p_j.

As shown in Figure 6, a tree shows which peer forwards messages to which peers. Here, a parent node p_i shows a relay peer which forwards values to the child peers on receipt of the values. A collection of the child peers shows a set $MPR(p_i)$ of relay peers of p_i. Peers colored black and white show relay and leaf peers, respectively. A subnetwork covered by a peer p_i is also a *subtree* of p_i. A peer which is chosen as a relay peer plays a significant role in the value exchange process. If a relay peer p is faulty, every peer covered by the peer p is not able to receive messages which are sent to the peer p. Let us consider a subtree S of a peer p shown in Figure 7, which is circled by the line. A peer p is a root of the subtree S. Suppose the peer p is faulty. Here, every peer in the subtree S cannot receive messages sent to the peer p. Thus, if a relay peer p_i is faulty, every peer in a subtree of p_i cannot receive messages.

In order to improve the robustness for broadcasting messages, we newly introduce the trustworthiness of a neighbor peer. A *trustworthy* peer is a peer which is operational and does not send malicious messages. A peer p_i selects trustworthy neighbor peers as relay peers. Then, the peer p_i sends a message to the neighbor peers and only the trustworthy neighbor peer forwards the message to the neighbor peers. Suppose a second neighbor peer p_k in $N^2(p_i)$ has multiple first neighbor peers $p_{k1}, \ldots, p_{kl_k}$ in $N(p_i)$ which are parents of p_k. Hence, a neighbor peer p_{kh} which is the most trustworthy is selected as a relay peer, i.e. child peer of p_k. The peer p_{kh} has the highest possibility to forward a message from p_i to p_k.

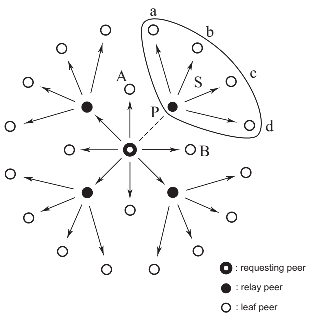

Fig. 7 Failure in Multipoint relays.

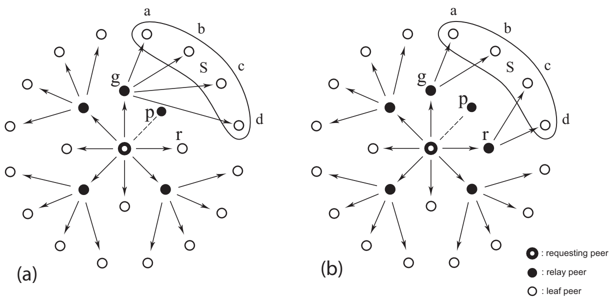

Fig. 8 Trusted neighbors in Multipoint relays.

Let us consider Figure 8 (a) as an example. Here, let $T(p_i)$ show the trustworthiness value of a peer p_i. In Figure 8, suppose $T(g) > T(r) > T(p)$ for three peers g, r, and p. Here, we select the most trustworthy one the peer g as a relay peer. Then, the peer g forwards message to every peer in the subtree S. This is a ideal case, that is, the subtree S which is originally covered by the peer p can be also covered by the peer g. However, the peer g might not be able to cover every peer in the subtree S as shown in Figure 8 (b). Therefore, another peer has to be selected to cover the peers which the peer g does not cover. In Figure 8 (b), the peers c and d uncovered by g are covered by the second most trustworthy peer r. The overall idea is that, every subtree is covered by a most trustworthy relay peer. It depends on overlay connections among peers how many number of relay peers are required to cover all the peers in a subtree. In Figure 8 (b), one more relay peer is required to cover the same subtree S as Figure 7. If we use more number of trustworthy neighbor peers to transmit messages to others, we can improve the overall fault-tolerancy of the multipoint-relay mechanism.

4. Trustworthiness of Peers

Differently from traditional centralized client-server systems, distributed systems are composed of multiple peers in a decentralized manner. This means, each peer has to obtain information of other peers and propagate the information to other peers through neighbor peers. A neighbor peer p_j of a peer p_i means that p_i can directly communicate with p_j. Thus, it is significant for each peer to have some number of neighbor peers. Moreover, it is more significant to discuss if each peer has trustworthy neighbor peers. In reality, each peer might be faulty or might send obsolete, even incorrect information to the other peers. If some peer p_j is faulty, other peers which receive incorrect information on the faulty peer p_j might reach a wrong decision. It is critical to discuss how a peer can trust each of its neighbor peers[23]. In this paper, we newly introduce a trustworthiness based multipoint relay algorithm by which the information can be move reliably broadcast every peer in the agreement procedure.

Suppose a requesting peer p_r would like to select a neighbor peer p_i as a relay peer for broadcasting a message with a package of values to the other peers. Let $T_r(p_i)$ show the trustworthiness of a neighbor peer p_i of a peer p_r, which the peer p_r holds. $N(p_r)$ shows a collection of neighbor peers of the requesting peer p_r. The peer p_r calculates the trustworthiness $T_r(p_i)$ for a neighbor peer p_i by collecting information on the peer p_i from every neighbor peer p_k in $N(p_r)$ which can communicate with both p_i and p_r, i.e. $p_k \in N(p_r) \cap N(p_i)$. There is some possibility that the peer p_i is faulty or sends malicious information. Hence, the peer p_r does not consider the information from the target peer p_i to calculate the trustworthiness $T_r(p_i)$.

A peer p_k sends a request to the peer p_i and receives a reply from p_i. This interaction is referred to as transaction. If p_k receives a successful reply in a transaction, the transaction is successful. Otherwise, it is unsuccessful. The peer p_k considers the neighbor peer p_i to be more trustworthy if p_k had more number of successful transactions for p_i. Let $Tv_k(p_i)$ indicate the *subjective* trustworthiness $T_k(p_i)$ on the target peer p_i which

Vol.2010-DPS-142 No.17
Vol.2010-CSEC-48 No.17
2010/3/4

a peer p_k obtains through communicating with the peer p_i. Let $tT_k(p_i)$ shows the total number of transactions which p_k issues to p_i. Let $sT_k(p_i)$ ($\leq tT_k(p_i)$) be the number of successful transactions from p_k to p_i. Here, the subjective trustworthiness $Tv_k(p_i)$ is calculated as follows:

$$Tv_k(p_i) = sT_k(p_i) \frac{t}{T}_k(p_i) \quad (1)$$

If the peer p_i is not a neighbor peer p_k, $p_i \in N(p_k)$, the peer p_k cannot obtain the subjective trustworthiness $Tv_k(p_i)$. In addition, if the peer p_k had not issued any transection to the peer p_i even if $p_i \in N(p_k)$, i.e. $tT_k(p_i)$ = 0, the subjective trustworthiness $Tv_k(p_i)$ is not defined. Here, $Tv_k(p_i)$ is assumed to be a "null" value. Thus, according to communication with each neighbor peer p_k, each peer p_r obtains the subject trustworthiness $Tv_k(p_i)$ for the neighbor peer p_i. The subject trustworthiness $Tv_k(p_i)$ shows how reliably a peer p_i is recognized by a peer p_k. Therefore, if a peer p_r would like to get the trustworthiness of a target peer p_i, the peer p_r asks each neighbor peer p_k to send the subjective trustworthiness $Tv_k(p_i)$ of the peer p_i. Each neighbor peer p_k keeps in record of the subject trustworthiness $Tv_k(p_i)$ in the log. Here, let $Tv(p_r)$ be a collection of neighbor peers which send the subjective trustworthiness $Tv_k(p_i) \neq$ null. After collecting the subjective trustworthiness $Tv_k(p_i)$ of the target peer p_i from each neighbor peer p_k, the requesting peer p_r calculates the trustworthiness $T_r(p_i)$ of the peer p_i by the following formula:

$$T_r(p_i) = \sum_{p_k \in [Tv(p_r) - \{p_i\}]} Tv_k(p_i) \frac{|}{T} v(p_r) - \{p_i\}| \quad (2)$$

Let us consider Figure 9 as a example. Here, a requesting peer p_r would like to know the trustworthiness $T_r(p_i)$ of a neighbor peer p_i. The peer p_r has five neighbor peers, p_1, p_2, p_3, p_4, and p_i. Here, $N(p_r)$ = { p_1, p_2, p_3, p_4, p_i }. A collection of neighbor peers of the peer p_r which excludes the peer p_i is indicated by a collection $S = N(p_r)$ - $\{p_i\}$ = { p_1, p_2, p_3, p_4 }. Here, the requesting peer p_r requests each neighbor peer p_k in the neighbor set S to send the subjective trustworthiness $Tv_k(p_i)$ of the peer p_i (k = 1, 2, 3, 4). After receiving the subjective trustworthiness of the peer p_i from all the four neighbors in S, the peer p_r calculates the trustworthiness $T(p_i)$ of the peer p_i by using the formula (2), $T(p_i)$ = ($Tv_1(p_1)$ + $Tv_2(p_2)$ + $Tv_3(p_3)$ + $Tv_4(p_4)$) / 4

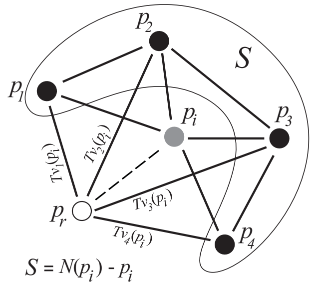

Fig. 9 Trustworthiness of peer.

By using the trustworthiness of each neighbor peer, the original multipoint relay (MPR) selection algorithm to select relay peers of p_i can be modified as follows:

1. Start with an empty multipoint relay set $MPR(p_i)$, $MPR(p_i) = \phi$. $S = N^2(p_i)$, $F = N(p_i)$. Let TF be a set of trustworthy neighbors, i.e. $\{p_j \in N(p_i) \mid T_r(p_j) \geq \alpha\}$ where $0 \leq \alpha \leq 1$. α gives a threshold value on the trustworthiness. If $T_r(p_i)$ is larger than or equal to α, the peer p_r recognized p_i to be trustworthy. Otherwise, p_i is considered to be untrustworthy.
2. While $TF \neq \phi$,
   (a) select a trustworthy neighbor peer p_i in TF such that $N(p_i) \cap N(p_j) = \phi$ for every trustworthy peer p_j in TF ($p_j \neq p_i$).
   (b) if found, $F = F$ - $\{p_i\}$, $TF = TF$ -$\{p_j\}$, $S = S$ - $N(p_i)$, $MPR(p_i) = MPR(p_i) \cup \{p_i\}$.
3. While $TF \neq \phi$,
   (a) select a trustworthy neighbor peer p_i in TF such that $|N(p_i) \cap S|$ is the maximum, i.e. the number of neighbor peers which are not yet covered is the maximum.
   (b) $F = F$ - $\{p_i\}$, $TF = TF$ - $\{p_i\}$, $S = S$ - $N(p_i)$, $MPR(p_i) = MPR(p_i) \cup \{p_i\}$, $N(p_i) = N(p_i) \cap S$.

5. Concluding Remarks

We discussed a flexible and efficient type of agreement protocol for a group of multiple peers where there is no centralized coordinator. Each peer is autonomous and makes a decision through directly communicating with the other peers. In order to efficiently make an agreement, we discussed the multi-value exchange (MVE) scheme where each peer sends a package of multiple possible values at each round. By using the MVE scheme, a group of multiple peers can easily and efficiently make an agreement. In the agreement procedure, each peer has to broadcast a package of multiple values to every peer in a group. We introduced the trustworthiness concept of neighbor peers. By using the trustworthy peer, we discussed a reliable and efficient way to broadcast values in a group of peers.

Acknowledgments

This research was partially supported by the strategy research project of Seikei University and MEXT, Grant in Aid for Building Strategy Research Infrastructure.

References

1) Aikebaier, A., Enokido, T., Takizawa, M.: Checkpointing in a Distributed Coordination Protocol for Multiple Peer Processes. *In: Proc. of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008)*, pp. 48–54. 2008
2) Aikebaier, A., Hayashibara, N., Enokido, T., Takizawa, M.: A Distributed Coordination Protocol for a Heterogeneous Group of Peer Processes. *In: Proc. of the IEEE 21th Conference on Advanced Information Networking and Applications(AINA 2007)*, pp. 565–572. 2007
3) Aikebaier, A., Hayashibara, N., Enokido, T., Takizawa, M.: Making an Agreement in an Order-Heterogeneous Group by using a Distributed Coordination Protocol. *In: Proc. of the 2nd International Workshop on Advanced Distributed and Parallel Network Applications (ADPNA 2007)*, CD-ROM. 2007
4) Aikebaier, A., Enokido, T., Takizawa, M.: A Distributed Coordination Protocol for Multiple Peer Processes. *In: Proc. of IEEE the 22nd International Conference on Advanced Information Networking and Applications (AINA 2008)*, CD-ROM. 2008
5) Aikebaier, A., Barolli, V., Enokido, T., Takizawa, M.: Recoverable Cuts to Make Agreement among Peers. *In: Proc. of IEEE the 23nd International Conference on Advanced Information Networking and Applications (AINA-2009)*, CD-ROM. 2009
6) Aikebaier, A., Enokido, T., Takizawa, M.: Efficiently Making Agreement among Peer Processes by using Recoverable Cuts. *In: Proc. of the 3rd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2009)*, CD-ROM. 2009
7) Corman, A. B., Schachte, P., Teague, V.: A Secure Group Agreement (SGA) Protocol for Peer-to-Peer Applications. *In: Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, pp. 24–29. 2007
8) Ezhilchelvan, P., Morgan, G.: A Dependable Distributed Auction System: Architecture and an Implementation Framework. *In: Proc. of the IEEE 5th International Symposium on Autonomous Decentralized Systems (ISADS)*, pp. 3–7. 2001
9) Gray, J., Lamport, L.: Consensus on Transaction Commit. *ACM Transactions on Database Systems (TODS) archive*, vol. 31(1), pp. 133–160. 2006
10) Hayes, B.: Cloud computing. *Communications of the ACM*, (7):9.11, July 2008.
11) Hurfin, M., Raynal, M., Tronel, F., Macedo, R.: A General Framework to Solve Agreement Problems. *In: Proc. of the 18th IEEE Symposium on Reliable Distributed Systems (SRDS)*, pp. 56–65. 1999
12) Kling, R.: Cooperation, Coordination and Control in Computer-supported Work. *Communications of the ACM*, vol. 34(12), pp. 83–88. 1991
13) Lamport, L., Shostak, R., Pease, M.: The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, vol. 4(3), pp. 382–401. 1982
14) Lee, P., Lui, J., Yau, D.: Distributed Collaborative Key Agreement Protocols for Dynamic Peer Groups. *In: Proc. of the 10th IEEE International Conference on Network Protocols*, pp. 322–331. 2002
15) Montresor, A.: A robust protocol for building superpeer overlay topologies. *In: Proc. of the 4th International Conference on Peer-to-Peer Computing*, pp. 202–209. 2004
16) Napster website, *http://www.napster.com*
17) Qayyum, A., Viennot, L., Laouiti, A.: Multipoint relaying for flooding broadcast messages in mobile wireless networks. *In: Proc. of 35th Annual Hawaii International Conference on System Sciences*, pp. 3866– 3875. 2002
18) Ripeanu, M. and Foster, I.: Mapping Gnutella Network. *IEEE Internet Computing*, January/February, pp. 50-57. 2002
19) Sabater, J., Sierra, C.: Reputation and Social Network Analysis in Multi-agent Systems. *In: Proc. of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, part 1, pp. 475–482. 2002
20) Shimojo, I., Tachikawa, T., Takizawa, M.: M-ary Commitment Protocol with Partially Ordered Domain. *In: Proc. of the 8th International Conference on Database and Expert Systems Applications (DEXA)*, pp. 397–408. 1997
21) Skeen, D.: NonBlocking Commit Protocols. *Proc. of the ACMSIGMOD International Conference on Management of Data*, pp. 133–142. 1981
22) Upadrashta, Y., Vassileva, J., Grassmann, W.: Social Networks in Peer-to-Peer Systems. *In: Proc. of the 38th Hawaii International Conference on System Sciences(HICSS-38 2005)*, CD-ROM. 2005
23) Watanabe, K., Nakajima, Y., Enokido, T., Takizawa, M.: Ranking factors in peer-to-peer overlay networks. *ACM Transactions on Autonomuous and Adaptive Systems (TAAS)*, vol. 2(3), September 2007