Human-centric Computing and Information Sciences
a SpringerOpen Journal

RESEARCH Open Access

Trustworthy Group Making Algorithm in Distributed Systems

Ailixier Aikebaier[1*], Tomoya Enokido[2] and Makoto Takizawa[1]

* Correspondence: alisher.akber@computer.org
[1]Department of Computers and Information Science, Faculty of Science and Technology, Seikei University, 3-3-1 Kichijoji-kitamachi, Musashino-shi, Tokyo 180-8633, Japan
Full list of author information is available at the end of the article

Abstract

Information systems are being shifted to scalable architectures like Cloud and peer-to-peer (P2P) models. In this paper, we consider the P2P model as a fully distributed, scalable system different from centralized coordinated systems in Cloud and Grid systems. A P2P system is composed of peer processes (peers). Here, applications are realized by activities of peers and cooperations among multiple peers. In P2P systems, since there is no centralized coordination, each peer has to obtain information about other peers by itself. In the group cooperation, each group member peer has to be trustworthy so that malicious behavior of a member peer cannot effect overall outcome of the whole group. Here, it is important to consider the trustworthiness of each group member as a base of an agreement procedure in the distributed environment. The goal of a group and the way to archive the goal are decided by the group members. During the agreement procedure, opinions of member peers have to be collected in a group. Malicious and unexpected behaviors of member peers can negatively effect the output of a group. Hence, it is significant to discuss how to compose a group only by including more trustworthy peers. In this paper, by taking advantage of the trustworthiness concept of each peer, we propose a novel approach to composing a trustworthy group in the distributed agreement protocols.

1 Introduction

The group cooperation is one of the most important actions in our human society. Without group cooperation, it is difficult to achieve any objective. It has been proven that cooperations among individual computers (peers) as a group are also really important in computer systems [1-3], like database transactions [4,5], robot technologies [6], and sensor-actuator networks [7]. Nowadays information systems are being shifted to distributed architectures from traditional centralized architectures. Peer-to-peer (P2P) systems are open world systems differently from other systems like the cloud computing model [8-10]. A huge number of computers and various types of computers with P2P applications are interconnected in large-scale P2P overlay networks lying on the top of underlying physical computer networks like the Internet Protocol (IP) networks. Differently from centralized or hybrid P2P systems, there is no centralized index server which manages the whole P2P system. Peers represents individual computers in the P2P system and autonomously take actions and cooperate with each other to realize the objective such as file sharing, building distributed storage, instant messaging, realizing distributed computation, contents delivery, and cooperative work. Because of the nature of the P2P systems, it is difficult for every peer to figure out what kinds of information are distributed to what peers, what kinds of peers exist in P2P overlay networks, and what kinds of relations among peers exist. In addition, malicious peers and faulty peers like crash-faulty peers can join and leave a P2P system without being authenticated and authorized. This causes a question on how each peer to trust a target peer in the P2P systems. In P2P applications like Intelligent Decision Advisor (IDA), Distributed Decision Making (DDM), and Computer Supported Cooperative Work (CSCW) [11,12], a group of multiple peers are required to do cooperation to realize some objective, for example, to fix a date of a meeting and to find a best location to build a building. Each member peer of the group plays an equally important role so that malicious and faulty behaviors of a peer can negatively effect the final output of the group. We introduce the trustworthiness concept of a peer [13], i.e. the more successfully a peer forwards messages, the more trustworthy the peer is. By taking advantage of trustworthiness concept [14] of peers, we propose a novel approach to creating a trustworthy group among peers.

In group communications [15,16], each peer has to deliver messages to every peer and receives messages from every peer in a group. There are many discussions on how to causally deliver messages in a group [17]. Efficient and reliable mechanisms to broadcast messages to every peer are required in order to casually deliver messages and realize the cooperation of multiple peers in a scalable group. The basic approach to broadcasting messages is the flooding algorithm [18]. Here, each peer sends a message to its neighbors and the neighbors forward the messages to their neighbor neighbor peers. In the multipoint relying (MPR) mechanism [19], each peer transmits a message to every neighbor peer but only some, not all of the neighbor peers forward the message. In order to increase the fault-tolerance, we discuss a novel *trustworthiness-based broadcast* (*TBB*) algorithm to reliably and efficiently deliver messages to every peer in a group. Here, each peer sends a message to its neighbor peers and only trustworthy peers out of the neighbor peers forward the message to their neighbors. Hence, even if untrustworthy peers are faulty, other peers can receive messages through trustworthy peers.

In section 2, we discuss the trustworthiness of peer and calculation of trustworthiness. In section 3, we present how to make a trustworthy group according to the trustworthiness of peers. In section 4, based on the trustworthy group concept we discuss trustworthiness-base broadcast (TBB) algorithm.

2 Trustworthiness of Peers

In P2P systems, each peer has to obtain information of other peers and propagate the information to other peers through neighbor (acquaintance) 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 p_i to have some number of neighbor peers. Moreover, it is more significant to discuss if each p_i can trust neighbor peers. In reality, each peer might be faulty. If some peer p_j is faulty, other peers might not be able to communicate with neighbor peers of the peer p_j. Hence, it is critical to discuss how a peer can trust each of its neighbor peers.

Let p_r be a peer with neighbor peers as shown in Figure 1. We would like to discuss the trustworthiness of each neighbor peer p_i of the peer pr. Let $T_r(p_i)$ show the

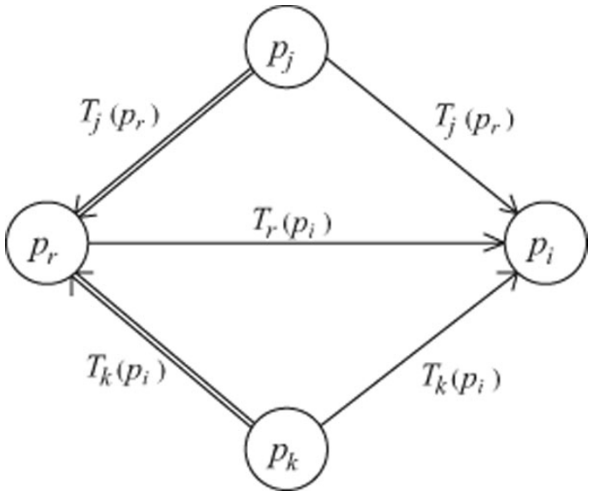

Figure 1 Trustworthiness of peer.

trustworthiness of a neighbor peer p_i of the peer p_r, which the peer p_r holds. $N(p_r)$ shows a collection of neighbor peers of the peer p_r. The peer p_r calculates the trustworthiness $T_r(p_i)$ for each neighbor peer p_i by collecting the trustworthiness values $T_k(p_i)$ 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 incorrect information. Hence, the peer p_r does not consider the trustworthiness $T_i(p_i)$ 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 request-reply interaction is referred to as *transaction*. If the peer p_k receives a successful reply from p_i, 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 issued more number of successful transactions for p_i. Let $ST_k(p_i)$ indicate the *subjective* trustworthiness $T_k(p_i)$ on the target peer p_i which a peer p_k obtains through directly communicating with the peer p_i. Let $tT_k(p_i)$ show the total number of transactions which the peer p_k issues to p_i. Let $sT_k(p_i)$ ($\leq tT_k(p_i)$) be the number of successful transactions which the peer p_k issues to p_i. Here, the subjective trustworthiness $ST_k(p_i)$ is calculated as follows:

$$ST_k(p_i) = \frac{sT_k(p_i)}{tT_k(p_i)} \tag{1}$$

If the peer p_i is not a neighbor peer of a peer p_k, $p_i \notin N(p_k)$, the peer p_k does not obtain the subjective trustworthiness $ST_k(p_i)$. In addition, if the peer p_k had not issued any transaction to the peer p_i even if $p_i \in N(p_k)$, i.e. $tT_k(p_i) = 0$, the subjective trustworthiness $ST_k(p_i)$ is not defined. Here, the subjective trustworthiness $ST_k(p_i)$ is assumed to be a "null" value. Thus, through communicating with each neighbor peer p_k, each peer p_r obtains the subject trustworthiness $ST_k(p_i)$ for the neighbor peer p_i. The subjective trustworthiness $ST_k(p_i)$ 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 $ST_k(p_i)$ of the peer p_i. Each neighbor peer p_k keeps in record the subjective trustworthiness $ST_k(p_i)$ in the log. Here, let $T\,N(p_r)$ be a collection of neighbor peers which send the

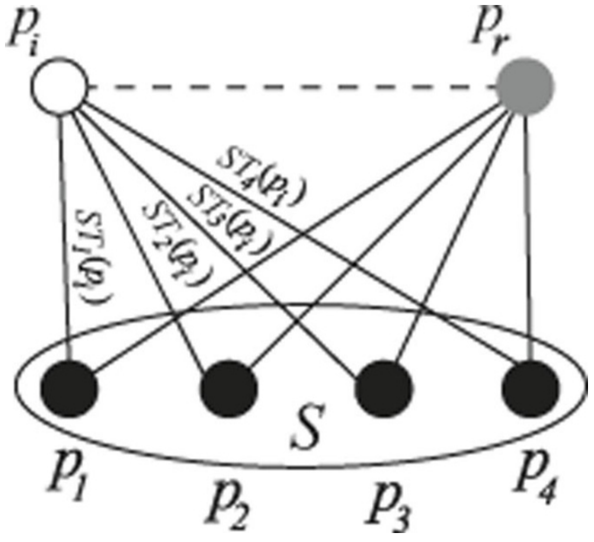

Figure 2 Subjective trustworthiness.

non-null subjective trustworthiness $ST_k(p_i)$ to the peer p_r. After collecting the subjective trustworthiness $ST_k(p_i)$ on the target peer p_i from every neighbor peer p_k, the source peer p_r calculates the trustworthiness $T_r(p_i)$ on the neighbor peer p_i by calculating the average value of the subjective trustworthiness values:

$$T_r(p_i) = \frac{\sum_{p_k \in TN(p_r)-\{p_i\}} ST_k(p_i)}{|TN(p_r) - \{p_i\}|} \tag{2}$$

Let us consider peers shown in Figure 2 as an example. Here, a source 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, $N(p_r) = \{p_1, p_2, p_3, p_4, p_i\}$. The peer p_i is excluded from $N\ (p_r)$ since p_i is a target peer, i.e. $S = N\ (p_r)$ - $\{p_i\} = \{p_1, p_2, p_3, p_4\}$. Here, the source peer p_r requests each neighbor peer p_k in the neighbor set S to send the subjective trustworthiness $ST_k(p_i)$ 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 the neighbor set S, the peer p_r calculates the trustworthiness $T_r(p_i)$ of the peer p_i by using the formula (2), i.e. $T_r(p_i) = (ST_1(p_i) + ST_2(p_i) + ST_3(p_i) + ST_4(p_i))$ / 4.

Now, let us consider three peers p_r, p_i, and p_j. Here, p_i is a neighbor peer of p_r and p_j is a neighbor peer of p_i while p_j is not a neighbor peer of p_r as shown in Figure 3. Through communicating with the neighbor peer p_i, the peer p_r obtains the trustworthiness $T_i(p_j)$ on the peer p_j. Here, we have to discuss how much the peer p_r can trust the non-neighbor peer p_j. In this paper, the *transitive* trustworthiness $TT_r(p_i)$ on a peer p_j is defined as follows:

$$TT_r(p_j) = T_r(p_i) \cdot T_i(p_j). \tag{3}$$

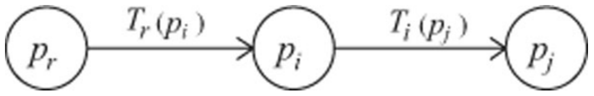

Figure 3 Transitive trustworthiness.

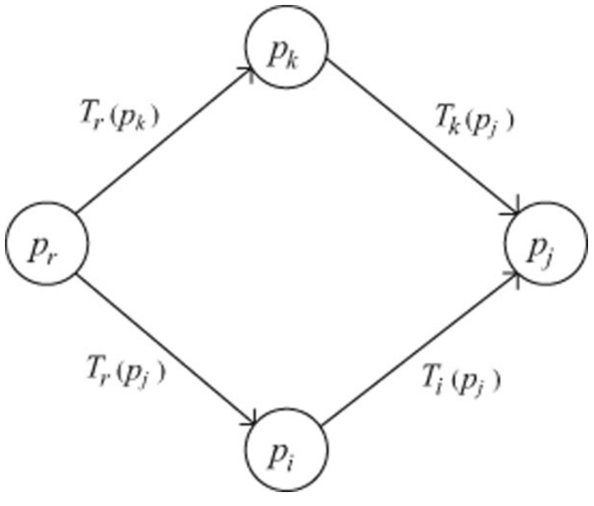

Figure 4 Transitive trustworthiness of peer.

Next, let us consider four peers shown in Figure 4. Here, a peer p_r has a pair of neighbor peers p_i and p_k which are neighbor of a target peer p_j. The transitive trustworthiness $T_r(p_k) \cdot T_k(p_j)$ and $T_r(p_i) \cdot T_i(p_j)$ might be different. In this paper, we calculate the transitive trustworthiness $TT_r(p_j)$ as follows.

$$TT_r(p_j) = \begin{cases} T_r(p_j) & \text{if } p_j \text{ is a neighbor of } p_r. \\ T_r(p_i) \cdot TT_i(p_j) & \text{if the condition } \alpha \text{ holds.} \end{cases} \quad (4)$$

Condition α: p_j is not a neighbor of p_r, p_i is a neighbor of p_r, and $T_r(p_i)$ is the maximum out of every neighbor of p_r where $TT_i(p_j)$ is defined.

3 Trustworthy Groups

3.1 Basic ideas

During distributed agreement procedures, first of all, the initiator peer p_i proposes an objective of a group G and invites others to the group G to do cooperation together with them. The initiator peer p_i sends an invitation message to its directly connected neighbor peers. Through the neighbor peers, the initiator peer p_i is connected with other peers and the group G of the peers is established. In this paper, the term “group” stands for the decision making committee which includes number of peers as members of the group. Each group makes decision on the given objectives by exchanging their opinions among group members.

In the previous works [20,21], we mainly discuss how to reliably deliver messages in a group of multiple peers after the group has been established. A group is constructed in a way that first neighbors, i.e. neighbors of an initiator peer are first included and then first neighbors of each first neighbor peer are included, until the number of members satisfy the group objectives like the scale of a required group. We discuss the trustworthiness-based broadcast (TBB) algorithm [22] to chose most trustworthy members to deliver the initiator message to the other peers as a relay peer in the group established. The trustworthiness of each peer is not considered when a group is established. The evaluation results o the TBB algorithm shows that, if peers in the group do not have enough number of directly connected neighbor peers, it is difficult to deliver messages to each peers in the group. The basic idea of the TBB algorithm is to chose most trustworthy peers (relay peer) to deliver messages to the other peers which do not have direct connections with the initiator peer. Since the relay peers forward the messages to other peers, the relay peers have to be more trustworthy. From the evaluation results, we found, if some peers which are selected as relay peers do not have enough number of first neighbor pees in the group, there is possibility that relay peers are not able to deliver the message from the initiator peer to all the other peers in the group. Here, some peers which are introduced to the initiator peer may not be trustworthy. That is, even if the peers receive messages, the peers may not forward the message to other peers. In this paper, we try to make a trustworthy group which is composed of trustworthy peers.

In this paper, we consider how to improve the trustworthiness of a group by including trustworthy peers in the group. If the group we call a decision making committee can be formed by more trustworthy peers from the beginning, we can significantly improve the reliability and efficiency of the whole decision making process afterward. We would like to discuss how to compose a group G so that every peer can receive messages in presence of untrustworthy peers. In P2P systems, an initiator peer which would like to make a group has to invite peers which the peer knows, i.e. neighbor peers. Then, the initiator peer invites its neighbors to the group.

The basic idea to make a trustworthy group G is that each peer only invites its trusted neighbor peers into the group G. Since an initiator peer p_i does not have enough number of neighbor peers to make a group, the initiator peer p_i asks its trustworthy neighbor peer p_j to introduce their neighbor peers to the initiator peer p_i. By choosing trustworthy peers among neighbor peers and introducing the trusted neighbor peers to the initiator peer p_i, only trustworthy member peers are included in the group G. There is smaller possibility the member peers who play a role of relay peer might be faulty.

3.2 Scale of a group

At the beginning stage of an agreement procedure, according to the objectives which the group aims at achieving, the scale of the group is decided. For example, more or fewer number of peers are required to be included in a group for different objectives. In the scientific computation, huge number of peers are required to be involved in the computation process and offer their computation power. In another case like schedule making or decision making in a group, only small number of peers may be required to be involved. But in either case, by selecting group members according to their behaviors in the history, we can somehow guarantee the future behaviors of the peers.

3.3 Creation of a trustworthy group

We assume each peer dynamically updates the subjective trustworthiness value of each neighbor peer on completion of each transaction with the corresponding neighbor peer. We also assume that each peer periodically calculates the trustworthiness value for each of its neighbor peer by requesting other neighbor peers to send the subject trustworthiness values. Therefore, each peer holds an up-to-date subjective trustworthiness value and trustworthiness value to each of its neighbor peers.

At first, the initiator peer p_o selects the most trustworthy peer which satisfies the trustworthiness requirement from its first neighbor peers depending on the trustworthiness record the initiator peer has on the neighbor peers. If the selected trustworthy peers from the first neighbors do not satisfy the scale of the group and more number of peers are required in the network, the initiator peer p_o requests the selected peers to become a relay peer and to introduce trustworthy peers from its neighbor peer p_j to the initiator peer p_o. Here, suppose the initiator peer p_o is introduced a peer p_j from a neighbor peer p_i. If $T_o(p_i) \cdot TT_i(p_j)$ is larger than some value, the initiator peer p_o takes the peer p_j as a relay peer. By repeating this procedure, enough number of trustworthy peers can be selected as the group members and a trustworthy group is created.

As shown in Figure 5, the initiator peer p_0 in the middle (triangle shape) asks only trustworthy neighbor peers p_{01}, p_{02}, p_{03}, p_{04} and p_{05} to make a group G. The black colored peers stand for the trustworthy peers to the initiator p_0 and white colored peers indicate untrustworthy peers. If peers p_{01}, p_{02}, p_{03}, p_{04} and p_{05} accept the invitation from the initiator peer p_0, the peers send acknowledgments to the initiator peer p_0 and are included in the group G. At this point, the initiator peer p_0 checks for the number of peers in the group G. If more number of peers are needed to be included in the group G, the initiator peer p_0 asks trustworthy neighbor peers p_{01}, p_{02}, p_{03}, p_{04} and p_{05} to introduce their trustworthy neighbors to p_o. As shown in Figure 5, the peer p_{01} introduces peers p_{11}, p_{12}, and p_{13} to the initiator peer p_o. Here, $T_{01}(p_{1i})$ is larger than the trustworthiness requirement T_{req}. The peer p_o takes every peer p_{1i} since T_0

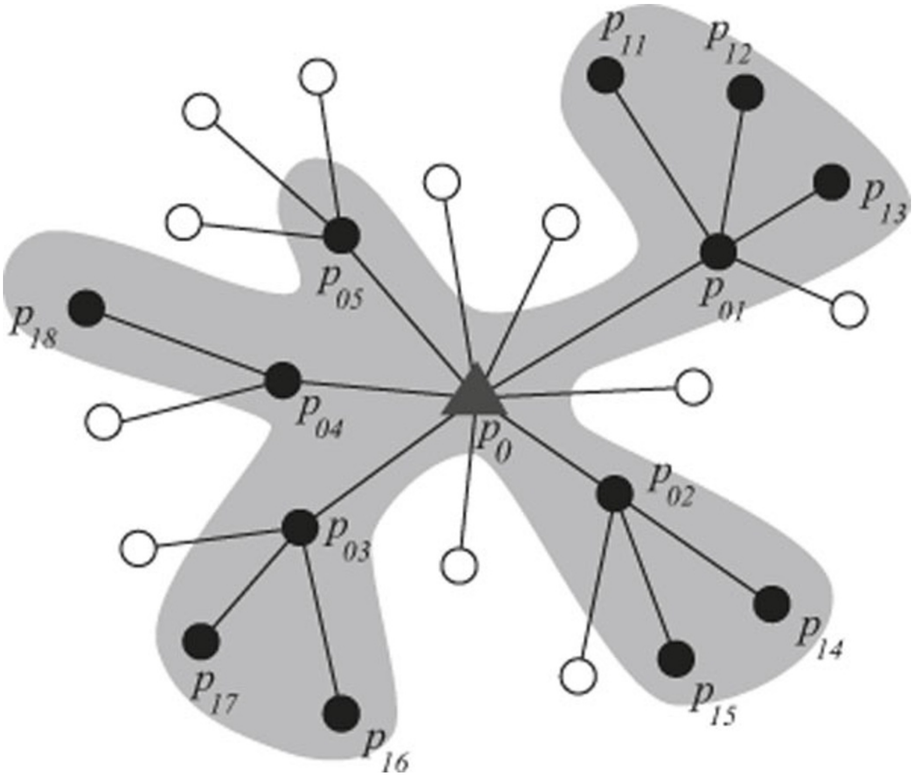

Figure 5 Group members selection.

$(p0_1) \cdot T_{01}(p(_{1i})) \geq$ for $i = 1,\ldots, 3$. The peer p_{02} introduces peers p_{14} and p_{15}. The peer p_{03} introduces peers p_{16} and p_{17}. The peer p_{04} introduces a peer p_{18} to the initiator peer p_0. Since the peer p_{05} does not have trustworthy neighbor peers which satisfy the trustworthiness requirement of the group G, no peer is introduced from the peer p_{05} to the initiator peer p_0. The initiator peer p_o invites the peers $p_{11},\ldots, p_{18}$ to the group G and the number of peers in the group satisfy the scale of the group G. Thus, the group G includes fourteen peers.

To create a trustworthy group, the following steps are taken:

1. The initiator peer p_0 decides on the scale S of the group G and the trustworthiness requirement T_{req}.

2. The initiator peer p_0 selects most trustworthy neighbors which satisfy the trustworthiness requirement ($\geq T_{req}$) as group members.

3. If the initiator peer p_0 could find enough number of peers ($\geq S$) among its neighbors, the group is successfully created.

4. If the initiator peer p_0 could not find enough number of group members ($\geq S$) from its neighbors, p_i asks selected trustworthy neighbors to introduce trustworthy neighbor peers.

5. If a selected peer introduces its trustworthy neighbor peers to the initiator peer p_0, the initiator peer p_0 invites every introduced peer which satisfies the trustworthiness requirement in the group. If the peer agree on member of the group G, the per is included in the group G. This step is repeated until the number of peers in the group get the group scale S.

6. Unless enough number of trustworthy peers could be found, the procedure terminates and the group creation fails.

By applying the trustworthiness concept into the group creation procedure, we can increase the reliability of the group. Only trustworthy peers are invited to the group. This means that there is smaller possibility that some member peer is faulty to broadcast messages to every member peer and the fault-tolerance of the group can be increased. On the other hand, groups where the trustworthiness concept of peers is not considered can be vulnerable to the network failure.

4 Trustworthiness-based Broadcast (TBB) Scheme

4.1 Multipoint relaying (MPR) scheme

In a group of multiple peers, each peer has to deliver a message to all the other peers. In a scalable P2P overlay network, each peer cannot directly send a message to every other peer of a group due to the scalability of the network. Each peer can only send a message to its neighbor peers, i.e. *acquaintance* peers. One approach to broadcasting a message is pure flooding scheme where messages are forwarded from peers to their neighbor peers. However, the pure flooding scheme implies the huge network overhead due to the message explosion.

The concept of "multipoint relaying (MPR)" scheme is developed to reduce the number of duplicate transmissions. 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 to other peers. Each peer is assumed to know not only the first neighbor peers but also the second neighbor peers. First neighbor peers are peers with which the peer can directly communicate. The peer is assumed to know every second neighbor peer, but cannot directly communicate with it. By taking into consideration the second neighbor peers, each peer selects a subset of the first neighbor peers only which forward the message. The selected neighbor peers are referred to as *relay* peers. The other neighbor peers which just receive the message and do not forward the message are *leaf* peers. In a *directed acyclic graph* (*DAG*) as shown in Figure 5, peers colored black and white to show relay and leaf peers, respectively. Relay peers (black one) forwards the message to the other peers, leaf peers (white one) only receives the message and does not forward it to the others. By reducing the number of peers to forward the message to the other peers, totally the MPR algorithm can significantly reduce the number of message which broadcast in the network. Therefore, we can save the network bandwidth for other network activities.

4.2 Message broadcasting

Normally, in order to broadcast a message from an initiator peer to every member peer in a group, the initiator peer sends the message to its neighbor peers. Then the neighbor peers forward the message to their neighbor peers and so on. Finally the message can be deliver to all members in the group.

To more reliably and efficiently broadcast messages to every peer in a group, we take into account the trustworthiness of each neighbor peer and newly introduce a way to deliver messages to the other members through most trustworthy neighbor peers. In our human society, we always consider the trustworthiness of a person as one of the most important factors to evaluate a person. We always would like to work with trustworthy persons. For example, if there is an important package we would like to deliver to someone and there is no way to directly deliver the package, we have to ask someone to deliver the package. In this case, we select a most trustworthy person to deliver the package, since there is smaller possibility a trustworthy person lose the package.

As discussed in the previous section, a group G is composed of trustworthy neighbors of the initiator peer p_i and trustworthy neighbors which are introduced to the initiator peer p_i as shown in Figure 6.

In Figure 6, there are 17 peers. We assume the trustworthiness requirement of a group G is $T_{req} \geq 5$ and the scale of the group $S = 10$. Since the trustworthiness requirement of the group G is $T_{req} \geq 5$, an initiator peer p_i only invites peers p_{01}, p_{02}, and p_{03} to the group G, because each of the peers may have a greater trustworthy value than T_{req}. The scale of the group $S = 10$ means that, the minimum number of trustworthy peers to compose a trustworthy group G is 10. The initiator peer p_i asks the selected peers p_{01}, p_{02}, and p_{03} to introduce their neighbor peers which have greater trustworthy values than T_{req}. On receipt of the request from the initiator peer p_i, the peer p_{01} only introduces its neighbor peer p_{10} to p_i, because the other peers cannot satisfy the trustworthiness requirement T_{req} of the group. In the neighbor peer p_{02}, none of its neighbor peers p_{t2}, p_{t3}, and p_{t4} can satisfy the trustworthiness requirement T_{req}. Thus, the peer p_{02} can introduce none of its neighbor to the initiator peer p_i. The peer p_{03} can introduce its neighbor peers p_{11} and p_{12} to the initiator peer p_i according to the trustworthiness requirement of the group G. Since the number of selected trustworthy peers still cannot satisfy the scale requirement S of the group G, the initiator peer p_i asks trustworthy peers p_{10}, p_{11}, p_{12}, and p_{13} newly included to introduce their trustworthy neighbor peers. Finally, the peer p_{12} introduces its neighbor

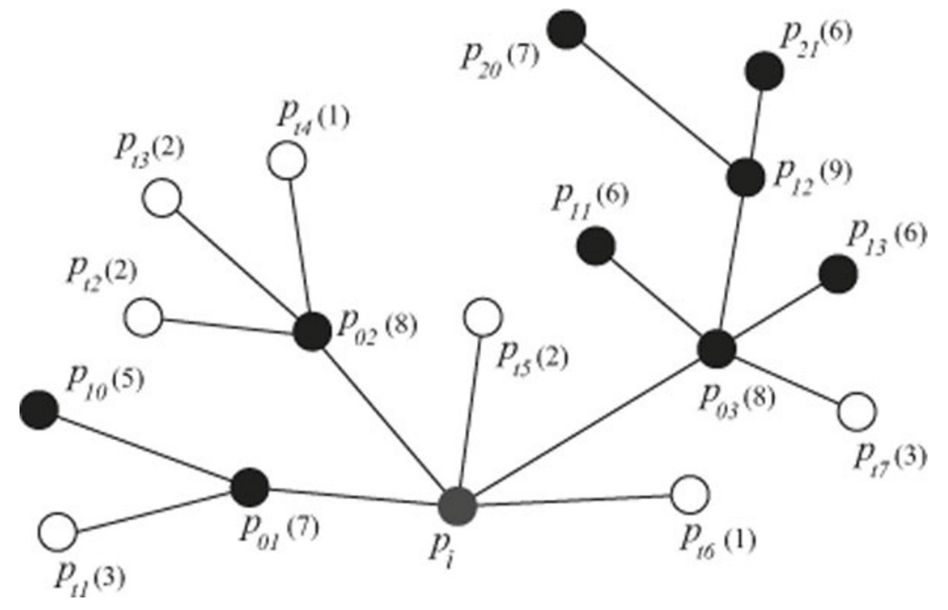

Figure 6 Introduction of neighbor peers.

peers p_{20} and p_{21} which satisfy the trustworthiness requirement T_{req} of the group G. Here, since the number of peers satisfy the scale requirement S of the group G, the group G is established and ready to do the group activities.

By including only the peers which satisfy the trustworthiness requirement T_{req} of the group G, the trustworthiness of the group can be guaranteed. Therefore, the initiator peer p_i knows about not only its directly connected neighbor peers but also other group members. Since other group members are introduced to the initiator peer p_i through neighbors of the initiator peer, the initiator peer knows which peer is introduced by which neighbor peer and the trustworthiness of the peers. The information about other members can be used by the initiator peer p_i to select effective and more reliable paths to broadcast messages.

The scenarios as shown in Figures 7, 8, and 9 indicate how an initiator peer p_i selects the message broadcast paths in order to more reliably and efficiently broadcast messages to every peer in the group G.

Based on the trustworthy group concept, we can increase the reliability of the message broadcasting procedure and fault tolerance of the group. In this paper, we also consider the efficiency of the message broadcasting procedure. That is, we have to reduce the number of messages to deliver messages to all the peers in a group G. In addition, by taking advantage of the TBB algorithm [22], we can increase the reliability of the message delivery process. According to the TBB algorithm, the most reliable path for a source peer to deliver messages to the other peers in the group G can be selected, even in presence of peer faults. Thus, messages can be delivered to all the peers in the group G.

Figures 7, 8, and 9 show some common scenarios showing how peers forward messages after a trustworthy group is established. The initiator peer p_i sends a message to its trustworthy neighbor peers p_{01} and p_{02} and then the peers forward the message to the peers p_{10},..., p_{15} as shown in Figure 7. Here, we discuss the scenarios shown in

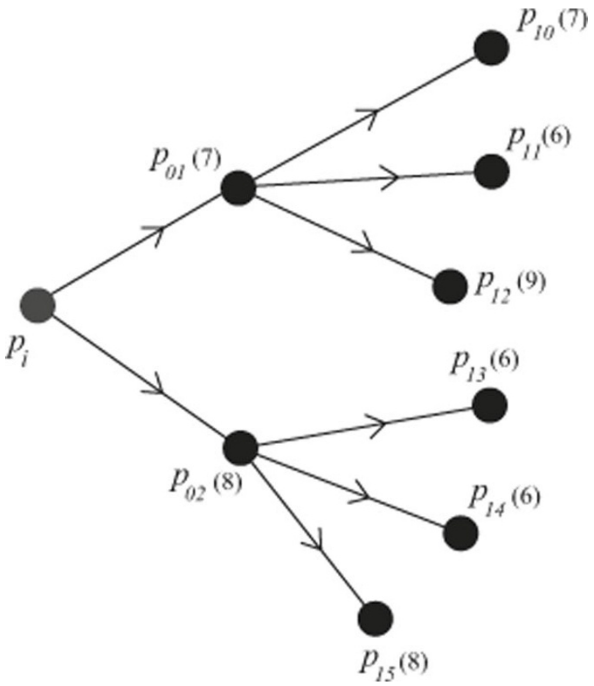

Figure 7 Message broadcast scenario 1.

Figures 8 and 9. Here, there is possibility that some peers are both neighbors of peers p_{01} and p_{02}. The peer p_{12} and peers p_{10}, p_{11}, and p_{12} are shared neighbor peers of both the peers p_{01} and p_{02}, respectively. Because at the group creation phase, the initiator peer p_i already has the information about each peer in the group G, e.g. the

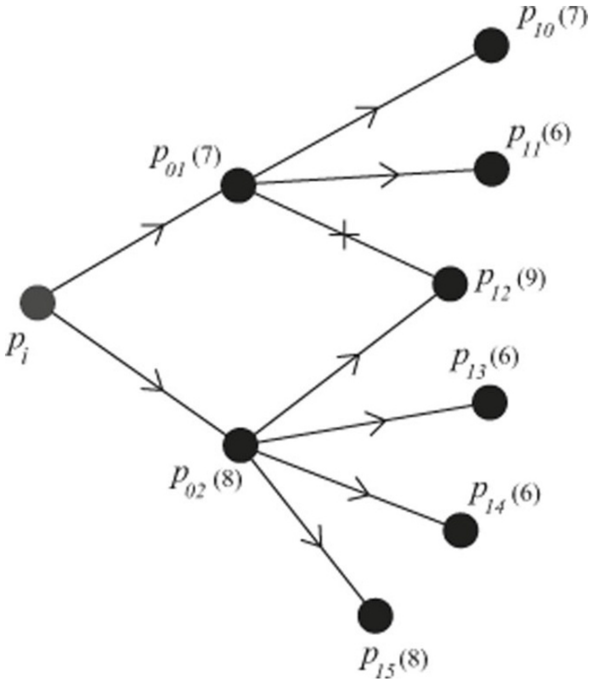

Figure 8 Message broadcast scenario 2.

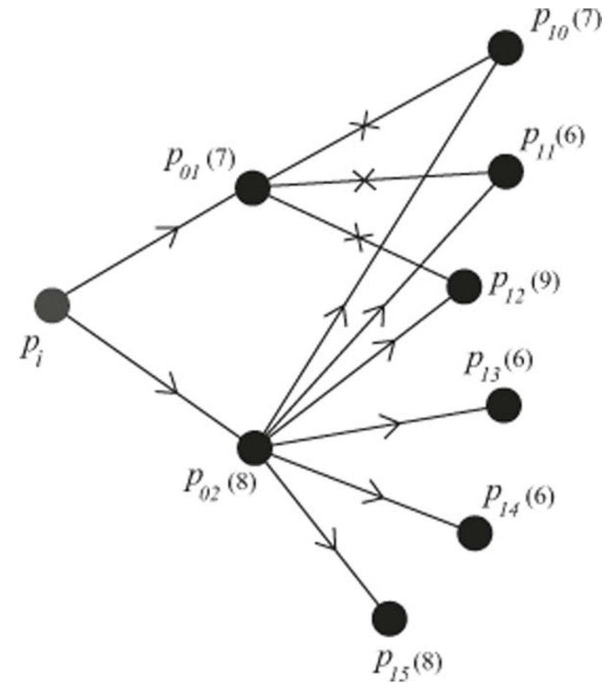

Figure 9 Message broadcast scenario 3.

trustworthiness value and so on. Therefore, the initiator peer p_i can select an efficient path to deliver messages to the other peers in the group G. For example, in Figure 9, since both peers p_{01} and p_{02} can forward messages to the peers p_{10}, p_{11}, and p_{12} and the initiator peer p_i knows about that. Since the peer p_{02} has a greater trustworthiness value 8 than the trustworthiness value 7 of the peer p_{01}, the initiator peer p_i selects the peer p_{02} to forward messages to the peers p_{10}, p_{11}, and p_{12}. The peer p_{01} does not forward messages to the peers p_{10}, p_{11}, and p_{12}. By applying this scheme, we can not only guarantee that messages can be more reliably delivered but also the number of unnecessary message delivery can be reduced in the network.

4.3 TBB algorithm

A relay peer plays a critical role to broadcast messages in a trustworthy group G. If a relay peer is faulty, every peer simply covered by the faulty relay peer is not able to receive messages. Since the group is composed by trustworthy peers, there is smaller possibility the trustworthy peers might be faulty. In addition, we modify our previous work, trustworthiness-based broadcast (TBB) algorithm based on the trustworthy group concept to furthermore increase the reliability and flexibility of message broadcasting procedure in the group G.

The depth D of a group G means how many times the relay peers have to forward a message to send the message from the initiator peer p_i to a member peer p_j of the group G. Let $P_{(D=h)}$ show collection of peers which can receive the message from the initiator peer p_i with h hops. As shown in Figure 10, since the depth D of peers p_{20}, p_{21}, and p_{22} is 3 (D = 3), $P_{(D\ =\ 3)}$ = {p_{20}, p_{21}, p_{22}}. Thus, the initiator peer p_i needs to deliver a message to peers in the set $P_{(D\ =\ 3)}$ through peers in $P_{(D\ =\ 2)}$ and $P_{(D\ =\ 1)}$.

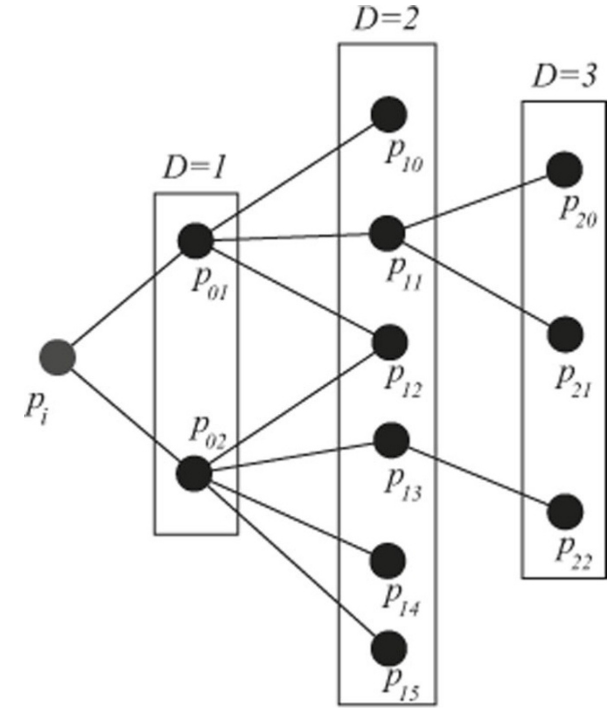

Figure 10 Depth of a group.

Relay peers in the set $P_{(D=i-1)}$ forward messages to peers in a set $P_{(D=i)}$. By checking peers in the sets $P_{(D=i)}$ and $P_{(D=i-1)}$, we can find whether or not some peers in the set $P_{(D=i)}$ receive message from multiple (≥ 2) relay peers in the set $P_{(D=i-1)}$. If a peer receives a message from multiple peers, we can select only the most trustworthy relay peer to deliver the message to the peer. Thus, we can not only more reliably deliver messages to peers but also reduce unnecessary message delivery.

For example, in Figure 11, we assume there are eight peers in the group G. The initiator peer p_i sends a message to other peers through a pair of directly connected trustworthy neighbor peers p_{01} and p_{02}. The peer p_{01} has three directly connected trustworthy neighbor peers p_{11}, p_{12}, and p_{13}. The peer p_{02} also has three directly connected trustworthy neighbor peers p_{13}, p_{14}, and p_{15}. A pair of peers p_{01} and p_{02} have a common trustworthy neighbor peer p_{13} so that both the peers p_{01} and p_{02} forward messages from the initiator peer p_i to the peer p_{13}. As defined, a set $P_{D=i}$ (i = 2) includes peers p_{11}, ..., p_{15} and a set $P_{D=i-1}$ (i = 2) includes peers p_{01} and p_{02}. By checking the sets $P_{D=i}$ and $P_{D=i-1}$, we can find the peers p_{01} and p_{02} forward messages to the peer p_{13}. Since the trustworthiness value of p_{01} is six and the trustworthiness value of p_{02} is eight as shown in Figure 11 so that a more trustworthy peer p_{02} is selected to forward messages to the peer p_{13}. The relay peer p_{01} would not forward messages to the peer p_{13}. By applying this algorithm to all peers in the group G, we can select a more trustworthy path to deliver messages to each peer in the group G and also reduce the unnecessary message delivery in the group G.

5 Concluding Remarks

In this paper, we discussed how to create a trustworthy group of multiple peers in a scalable P2P overlay network. In the decentralized scalable P2P networks, it is difficult

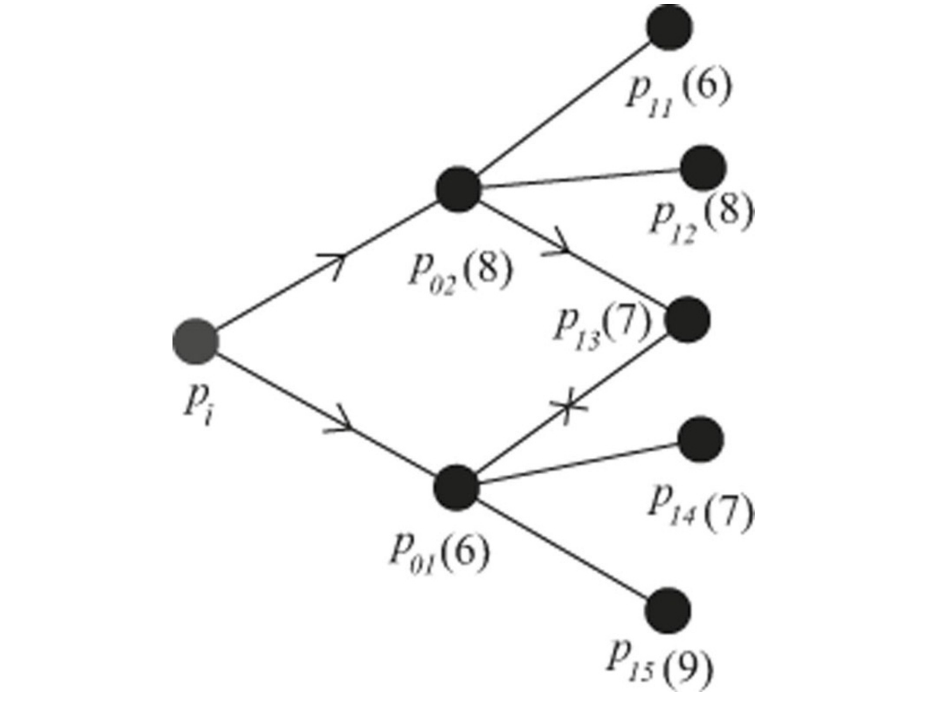

Figure 11 **Trust-based message broadcasting**.

to make sure the correctness of information. Only trustworthy neighbor peers of a peer can provide the peer with valid information. In a group, all group members must be trustworthy so that malicious action of a peer can not effect the whole group. Hence, only trustworthy neighbors are invited to make the group. By using the trustworthiness of peers, we newly proposed the trustworthy group concept where only trustworthy neighbor peers are included in the group. The reliability of a group and fault tolerance of message broadcasting procedure of agreement protocols are increased. We also discussed an efficient and reliable way to broadcast messages to all the peers in a trustworthy group. By taking advantage of the trustworthiness-based broadcast (TBB) algorithm, we newly introduced the algorithm to choose most reliable path to deliver message to all the peers in the trustworthy group. By the combinations of the trustworthy group concept and the TBB algorithm, not only messages can be more reliably delivered to all the peers in the group but also the number of unnecessary message delivery can be reduced in the network.

Acknowledgements
This research is supported by Research Fellowships of Japan Society for the Promotion of Science for Young Scientists (JSPS). This research was also partially supported by the strategy research project of Seikei University and MEXT, Grant in Aid for Building Strategy Research Infrastructure.

Author details
[1]Department of Computers and Information Science, Faculty of Science and Technology, Seikei University, 3-3-1 Kichijoji-kitamachi, Musashino-shi, Tokyo 180-8633, Japan [2]Faculty of Bussiness Administration, Rissho University, 4-2-16, Osaki, Shinagawa, Tokyo, 141-8602, Japan

Authors' contributions
Ailixier Aikebaier and Makoto Takizawa conceived the algorithm and analysed the experiment data together. Tomoya Enokido and Ailixier Aikebaier designed and performed the simulation and evaluations. Ailixier Aikebaier and Makoto Takizawa co-wrote the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests'

Received: 18 October 2011 Accepted: 22 November 2011 Published: 22 November 2011

References

1. Corman AB, Schachte P, Teague V (2007) A Secure Group Agreement (SGA) Protocol for Peer-to-Peer Applications. Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07) 24–29
2. Ezhilchelvan P, Morgan G (2001) A Dependable Distributed Auction System: Architecture and an Implementation Framework. Proc. of the IEEE 5th International Symposium on Autonomous Decentralized Systems (ISADS) 3–7
3. Gray J, Lamport L (2006) Consensus on Transaction Commit. ACM Transactions on Database Systems (TODS) archive 31(1):133–160
4. Taniar David, Wenny Rahayu J, Leung HCClement, Goel Sushant (2009) Advances in high performance database technology. Proceedings of the 11th International Conference on Information Integration and Web-based Applications & ServicesiiWAS
5. Taniar David, Leung HCClement, Wenny Rahayu J, Goel Sushant (2008) High Performance Parallel Database. Processing and Grid Databases John Wiley & Sons
6. Belta C, Kumar V (2004) Abstraction and control for Groups of robots. IEEE Transactions on Robotics 20(5):865–875
7. Waluyo AB, Taniar D, Srinivasan B, Rahayu JW, Takizawa M (2011) Adaptive and Efficient Data Dissemination in Mobile P2P Environments. The 25th IEEE International Conference on Advanced Information Networking and Applications Workshops (AINA-2011) 861–866
8. Foster I., *et al* (2008) Cloud Computing an Grid Computing 360-Degree Compared. Proc. IEEE Grid Computing Environments Workshop, IEEE Press 1–10
9. Armburst M., *et al* (2009) Above the Clouds: Berkeley View of Cloud Computing. tech report UCB/EECS-2009-28, Electrical Eng. and Computer Science Dept., Univ. of California, Berkeley
10. Hayes B (2008) Cloud computing. Communications of the ACM 51(7):9–11
11. Richardson T, Stafford-Fraser Q, Wood KR, Hopper A (1998) Virtual network computing. IEEE Internet Computing 2(1):33–38
12. Kling R (1991) Cooperation, Coordination and Control in Computer-supported Work. Communications of the ACM 34(12):83–88
13. Aikebaier A, Enokido T, Takizawa M (2010) Trustworthiness among Peer Processes in Distributed Agreement Protocol. Proc. of IEEE the 24nd International Conference on Advanced Information Networking and Applications (AINA 2010), CD-ROM
14. Watanabe K, Nakajima Y, Enokido T, Takizawa M (2007) Ranking factors in peer-to-peer overlay networks. ACM Transactions on Autonomuous and Adaptive Systems (TAAS) 2(3). Article No. 11
15. Lamport L (1978) Time, Clocks and the Ordering of Events in a Distributed System. Communications of the ACM 21 7:558–565
16. Chockler VGregory, Keidar Idit, Vitenberg Roman (2001) Group communication specifications: a comprehensive study. ACM Computing Surveys (CSUR) 33(4):427–469
17. Kawanami S, Enokido T, Takizawa M (2004) A Group Communication Protocol for Scalable Causal Ordering. Proc. of the 18th International Conference on Advanced Information Networking and Applications (AINA'04) 1:296–301
18. Ripeanu M, Foster I (2002) Mapping Gnutella Network. IEEE Internet Computing 50–57
19. Qayyum A, Viennot L, Laouiti A (2002) Multipoint relaying for flooding broadcast messages in mobile wireless networks. Proc. of the 35th Annual Hawaii International Conference on System Sciences 3866–3875
20. Aikebaier A, Hayashibara N, Enokido T, Takizawa M (2007) A Distributed Coordination Protocol for a Heterogeneous Group of Peer Processes. Proc. of the IEEE 21th Conference on Advanced Information Networking and Applications (AINA 2007) 565–572
21. Aikebaier A, Enokido T, Takizawa M (2008) A Distributed Coordination Protocol for Multiple Peer Processes. Proc. of IEEE the 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), CD-ROM
22. Aikebaier A, Enokido T, Takizawa M, Deen SM (2010) TBB-Scheme for Reliably Broadcast Messages among Peer Processes. Proc. of the 13th International Conference on Network-based Information Systems (NBiS2010) 337–344

doi:10.1186/2192-1962-1-6

Cite this article as: Aikebaier *et al.*: **Trustworthy Group Making Algorithm in Distributed Systems.** *Human-centric Computing and Information Sciences* 2011 **1**:6.