Proposal and Evaluation of a Network Construction Method for a Scalable P2P Video Conferencing System

Hideto Horiuchi, Naoki Wakamiya, and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University
1–5 Yamadaoka, Suita-shi, Osaka 565–0871, Japan
{h-horiuti,wakamiya,murata}@ist.osaka-u.ac.jp
http://www.anarg.jp

Abstract. Recently, video conferencing systems based on peer-to-peer (P2P) networking technology have been widely deployed. However, most of them can only support up to a dozen of participants. In this paper, we propose a novel method to construct and manage a P2P network for a scalable video conferencing system. Our method consists of three parts: a network construction mechanism, a tree reorganization mechanism, and a failure recovery mechanism. First, the network is constructed as new peers join a conference. Then, the tree topology is reorganized taking into account the heterogeneity of the available bandwidth among peers and their degree of participation so that, those participants, i.e., peers that can have many children peers and/or often speak are located near the root of the tree. As a result, the delay from speakers to the other participants is reduced. Through simulation experiments, we verify that our tree reorganization mechanism can offer smooth video conferencing.

Key words: video conferencing, P2P, scalability, tree construction

1 Introduction

With the proliferation of the Internet, video conferencing systems are getting widely accepted making it possible to have a meeting or a discussion among people at different and distant places. Especially, video conferencing systems using an application level multicast (ALM) technology based on P2P communication have been introduced due to their ease of deployment and low cost of operation [1] [2]. However, they still have the scalability problem and most of them can only support at most a dozen of participants. For example, a company with worldwide branches and convenience chain stores may involve hundreds of managers in a business meeting. There has been a number of research works in scalable ALM algorithms, but they mainly consider distribution type of applications [3] [4]. Therefore, we need a video conferencing system that can accommodate hundreds or thousands of interactive participants.

In this paper, we propose a novel method for constructing and managing a P2P network for a scalable video conferencing system. We assume that participants, i.e., peers, dynamically join and leave a conference. Peers are heterogeneous in terms of the network capacity available for video conferencing. Our proposed system consists of three parts: a *network construction mechanism*, a *tree reorganization mechanism*, and a *failure recovery mechanism.* The network construction mechanism sets up a hierarchical distribution network, which consists of distribution trees consisting of tens or hundreds of peers, and a core network which interconnects these trees with each other. To have a smooth conference, it is necessary to keep the delay from speakers to other participants small. To accomplish this goal, we focus on the fact that the number of simultaneous speakers is limited whereas speakers dynamically change in accordance with the agenda. The tree reorganization mechanism dynamically reorganizes a distribution tree so that speakers are located near the root in a distribution tree. In addition, to reduce the height of a distribution tree, the tree reorganization mechanism dynamically moves peers with higher available bandwidth toward the root. Furthermore, in the case of failure in distribution of conference data due to a halt or disappearance of a peer, the failure recovery mechanism reconfigures the distribution network through local interactions among peers using local information acquired during network construction.

The rest of this paper is organized as follows. We describe our proposal in Section 2. Then, we present some simulation results in Section 3. Finally, we summarize the paper and describe some future work in Section 4.

2 Network Construction Method for Scalable P2P Video Conferencing

In this section, we give an overview of the scalable P2P video conferencing system consisting of the network construction mechanism, the tree reorganization mechanism, and the failure recovery mechanism. In the following, we use the terms peer and participants interchangeably.

2.1 Overview of Scalable P2P Video Conferencing System

Our system consists of a login server, peers, and a distribution network. Delivery and exchange of streaming data, i.e., video and audio are done through the distribution network. For low bandwidth requirement and management cost, we adopt a shared-tree architecture to the distribution network. The distribution network consists of the core network and the distribution trees in which the root is connected to the core network. In this paper, we call a peer which belongs to the core network *leader peer*, and all other peers *general peers.* A leader peer manages the IP addresses of neighboring leader peers and all children peers that are directly connected to it. A general peer keeps the IP addresses of its parent and children, and the list of the IP addresses, which it knows, in its *ancestor list.* Peers have a limitation on the number of acceptable children called

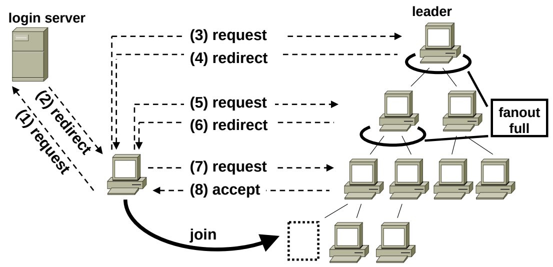


Fig. 1. Participation to a tree through sequential introduction

fanout in accordance to their available bandwidth. The login server is responsible for registration and management of the conference, and the authentication of participants. It manages only information of leader peers and the number of general peers in each tree, and not the structure of each tree.

The overview of the system behavior is as follows. First, a newly participating peer requests the login server for authentication. At this time, the participating peer is notified whether it should become a leader peer or general peer. Then, it connects to either the core network or a distribution tree to join the conference. Then, the participant is involved in the conference as a speaker or an audience in accordance with the agenda. Since we do not consider any management of speech coordination in this proposal, all participants can speak freely. Streaming data from a speaker is once transmitted to the root of the tree to which it belongs, and then broadcasted to the other peers in the tree and to peers in the other trees via the core network. Our method makes peers with high fanout, i.e., high bandwidth, and active speaking move to the root of tree. We call this *promotion*. The promotion reduces the tree height and delay between active peers and others. In video conferencing systems, peers may leave because of failures in routers or links. So our method dynamically recovers from the failure in the distribution network so that peers can continuously receive streaming data.

2.2 Network Construction Mechanism

In our method, a participating peer first gets authenticated by the login server and then connects to the distribution network. With consideration of the fanout of the peer and the number of peers in each tree, the login server determines the role of the peer. If a participating peer is determined as a leader peer, the peer gets the IP address list of other leader peers, measures delay to them, and connects to the neighbor peers.

If the participating peer is specified as a general peer, it connects to the designated distribution tree by sequential introduction of a temporary parent as shown in Fig. 1 [5]. First, the login server notifies the participating peer of the IP address of an appropriate leader peer as a temporary parent (Fig. 1:1-2). In our mechanism, the leader peer to be introduced is selected in a round-robin fashion. Therefore, without any peer leaving, the number of peers is equal among trees.

The participating peer deposits the notified IP address in its ancestor list and sends a participation request message to the temporary parent (Fig. 1:3). The temporary parent which receives the participation request message compares its fanout with the number of children. If its number of children is less than $fanout-1$, the temporary parent accepts the request and connects to the peer. The reason for comparing with $fanout-1$ is that the tree reorganization mechanism requires one spare link as will be explained later. On the other hand, if the number of children is equal to $fanout - 1$, the temporary parent introduces one of its direct children to the participating peer as a new temporary parent. We call this procedure *redirect* (Fig. 1:4). If the temporary parent has information about the topology of its descendants, by introducing a child with the lowest or smallest subtree, we can build a balanced tree. However, for this purpose, peers have to maintain the up-to-date information by exchanging control messages very often. Therefore, in our mechanism, we consider that a new temporary parent is selected among children in a round-robin fashion. We can expect that a distribution tree is constructed in breadth-first order and the delay from the leader peer can be reduced. The participating peer adds the IP address of the introduced temporary parent to its ancestor list and sends a participation request message (Fig. 1:5). By repeating these procedures, the participating peer can eventually connect with the temporary parent, which has an available link, and join the distribution tree (Fig. 1:6-8). The participating peer has all IP addresses of its ancestors in the ancestor list when connecting to the tree.

In this mechanism, there is only small overhead at the login server and peers, because no centralized unit manages the distribution tree topology and the additional load of processing messages occurs only at temporary parents. An additional advantage is that a peer can reconfigure the distribution tree during failure with only local interaction because a peer has the knowledge of the complete ancestor list.

2.3 Tree Reorganization Mechanism

In our method, a peer with high activity and high fanout moves to the root of the tree for low delay and smooth conferencing. We call it *promotion*. In addition, to reduce tree height by completing the fanout, a peer which has less than $fanout - 1$ children invites its grandchild as a direct child. In this section, we describe the details of this mechanism.

Peer Promotion A peer starts the promotion process if the participant speaks continuously. Additionally, a peer compares its fanout with that of its parent periodically. If the fanout is more than its parent, the peer starts the promotion process. However, the promotion process does not occur if the peer is involved in other tree reorganization or failure recovery. The promotion means that a peer becomes a child of its grandparent as shown in Fig. 2. Firstly, peer A which starts the promotion sends a promotion request message to parent peer B and its children (Fig. 2:1). If a peer receiving the request is involved in other tree

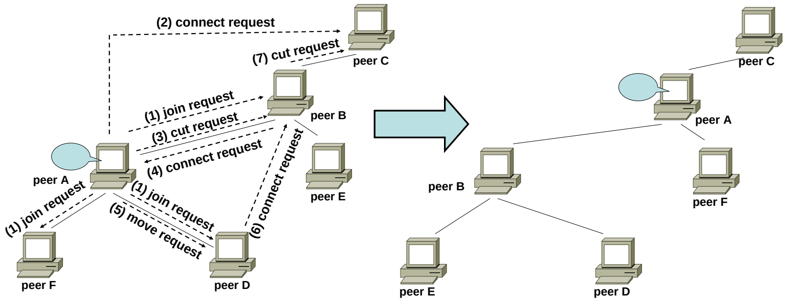


Fig. 2. Promotion of peer A for speaking

reorganization or failure recovery, it rejects the request, otherwise it sends back an accept message. The accept message from peer B has the IP address of its parent peer C. On receiving the accept message from all peers, peer A sends a connection request message to peer C (Fig. 2:2). If peer C is involved in other tree reorganization or failure recovery, peer C sends back a reject message, otherwise it makes a connection to peer A and sends back an accept message. If the number of children becomes equal to the fanout on peer C, the accept message from peer C includes information indicating that the spare link is used.

After connecting with peer C, peer A sends a disconnection request message to its previous parent B (Fig. 2:3). This request includes information whether the spare link of peer C is used. After receiving the request, peer B terminates the connection with peer A. If the spare link is not used on peer C, the promotion is completed at this time. Now, both peer A and B are children of peer C.

If the spare link is used on peer C, peer B, the previous parent of the promoted peer A, becomes a child of peer A to make one link free on peer C. First, peer B sends an adoption request message to peer A (Fig. 2:4). If the number of children is less than $fanout - 1$ on peer A, peer A accepts peer B as its child. On the other hand if equal, peer A sends a moving request message to peer D which is selected from peer A's children in a round-robin fashion to make a room for peer B (Fig. 2:5). The request message includes the IP address of peer B. Then peer D sends a connection request message to peer B (Fig. 2:6). Peer B accepts peer D as its child, peer D terminates the connection with peer A, and peer A becomes a parent of peer B. Then, peer B sends a disconnection request message to peer C (Fig. 2:7) and peer C terminates the connection. As a result, peer C obtains a new spare link. In this way, the promotion is completed.

Completing the Fanout Peers periodically compare the number of their children with the fanout. If the number is less than $fanout - 1$, a peer starts completing the fanout. However, if a peer is involved in other reorganization or failure recovery, the process does not occur. Peer A, which can accommodate more children, sends an introduction request message to peer B which is selected in a round-robin fashion from its children (Fig. 3:1). If peer B does not have any

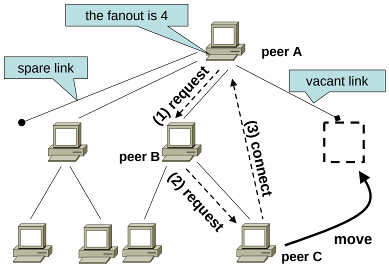


Fig. 3. Completing the fanout of peer A

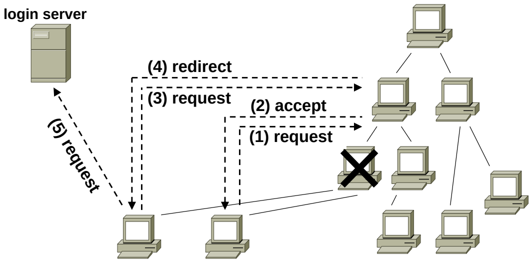


Fig. 4. Failure recovery

children, peer B sends back a reject message to peer A. Otherwise, peer B sends a moving request message to peer C which is selected in a round-robin fashion from its children (Fig. 3:2). The moving request includes the IP address of peer A. Peer C sends a connection request to peer A (Fig. 3:3) and makes the connection. After establishing the connection with peer A, peer C terminates the connection with peer B and this process is completed. If either of peer B and C or both are involved in other tree reorganization or failure recovery, peer A receives a reject message and the process is canceled.

2.4 Failure Recovery Mechanism

A peer may become to be unable to receive data due to not being able of accessing its temporary parent in tree construction/reorganization, a halt of links or routers, or a parent peer leaving the conference. We define this event as *failure*. In the failure recovery mechanism, a peer detecting a failure tries to make a new connection with another peer in its ancestor list [5].

If a peer fails in sending a message to a temporary parent, it sends a re-connection request message to the previous temporary parent, which introduced the missing temporary parent. On the other hand, if a peer detects the leaving or a fault of its parent, it sends a re-connection request message to its grandparent in the ancestor list (Fig. 4:1,3). In both cases, the IP address of the missing parent is removed from the ancestor list. If the recovering peer fails in sending the re-connection request message due to departure of the new temporary parent, it first removes the corresponding IP address from the ancestor list and then moves to the next ancestor at the bottom of the list. If the list becomes empty, the recovering peer goes to the login server and joins the distribution tree again as a new peer. (Fig. 4:5). On receiving the re-connection request message, the temporary parent establishes a connection with the recovering peer if the number of children is less than $fanout - 1$ (Fig. 4:2), or introduces a child to the recovering peer as a new temporary parent otherwise (Fig. 4:4). In the latter case, the requesting peer eventually joins the tree and reorganizes its ancestor list by the same process as the initial join.

If a child of a leader peer detects the failure of the leader peer, the peer notifies the login server of the failure. The login server appoints the peer which first sends the notification as a new leader peer and updates the information of leader peers. The other children of the missing leader peer also report the failure and are redirected to the new leader peer.

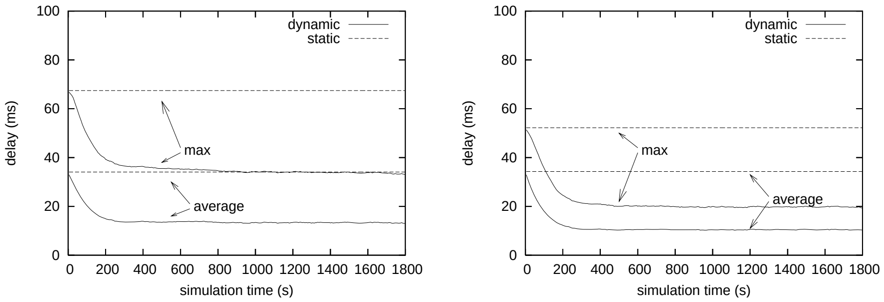
3 Simulation Experiments

In this section, we show simulation results to evaluate the tree reorganization mechanism which contributes to smooth video conferencing.

3.1 Simulation Conditions

In this paper, we focus on the performance and effectiveness of the tree reorganization mechanism and thus consider a single distribution tree in simulation experiments. Evaluation of the whole proposal is left as a future work. First, we create a physical network, which follows the power-law principle based on BA model [6] using BRITE [7]. The average degree of this network is 2, and it consists of 101 nodes, i.e., routers. Each router has one peer to participate in the conference. One peer of these serves as the login server. The fanout is fixed during the simulation and is equal to the degree of the designated router plus 1. The delay among peers is computed by physical hops over the shortest path by the Dijkstra method ignoring the access link between a peer and a router, and the propagation delay over one physical link is 1 msec. We do not consider transmission delay and processing delay. Peers participate in the conference at random time with uniform distribution from 0 to 10 seconds. The first participating peer becomes the leader peer. After 100 peers participate in the conference and construct the tree, no further peer joins and leaves the conference.

After all peers participate in the conference, a peer begins to speak and the tree reorganization is conducted. We call peers to speak as *candidates*. Ten candidates are randomly chosen at the start and are fixed during the simulation. The duration of each speech is exponentially distributed with a mean value of 6 seconds [8] and the minimum duration is 1 msec. Any one of the candidates is always speaking during the simulation. In other words, when a candidate stops speaking, the next speaker is randomly chosen among candidates and starts speaking



(a) Delay between the leader peer and all peers

(b) Delay between the leader peer and candidates

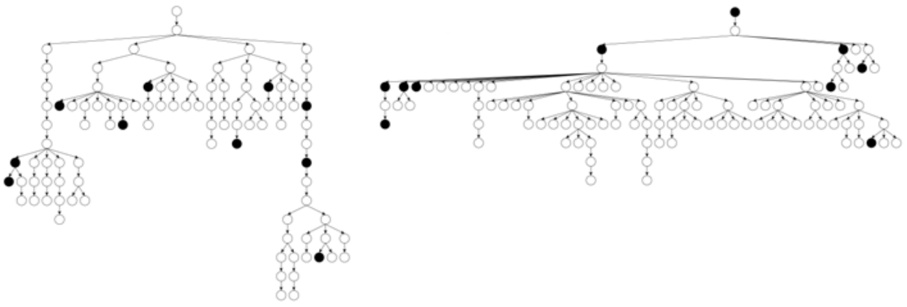
Fig. 5. The transition of delay

immediately. The same candidates would be chosen as the next speaker, but only one candidate speaks at the same time. In the following figures, time zero corresponds to the instant when the first speech starts. A speaking peer starts the promotion when it continuously speaks for more than 5 seconds, and as long as it is speaking, it tries the promotion every 5 seconds. However, as described in Section 2.3, if the preceding promotion is not completed, the next promotion is not triggered. All peers compare its fanout with its parent every 24 seconds and they may start the promotion depending on the result. To distribute the timing of the promotion among peers, the first comparison occurs at a random time with uniform distribution from 0 to 24 seconds. Peers compare the number of their children with their fanout every 7 seconds and they may start completing the fanout depending on the result. To distribute the timing, the first comparison occurs at random time with uniform distribution from 0 to 7 seconds.

We evaluate our method from the viewpoint of the average and maximum of delay from all peers to the leader peer and from all candidates to the leader peer, and the average and maximum number of received messages per peer. In the figures, we also show results of the case that a distribution tree does not change during a simulation experiment, denoted as *static*, to compare with results of the case with the tree reorganization mechanism, denoted as *dynamic*. Following results are the average over 1000 simulation experiments, each of which lasts for 30 minutes in simulation time unit after the first speaker begins to speak.

3.2 Simulation Results

Figure 5(a) illustrates the average and maximum delay between the leader peer and all peers. The figure shows that the tree reorganization mechanism can effectively reduce both of the average and maximum delay. Among promotions,



(a) The initial tree topology

(b) The reorganized tree topology

Fig. 6. Result of tree reorganization

those invoked by fanout comparison and completion mainly contribute to the initial reduction of delay. When the maximum delay between the leader peer and all peers in a distribution tree is D and that between leader peers in the core network is L, the maximum end-to-end delay among all peers can be derived as $D \times 2 + L$. Except for the initial stage, D is about 35 msecs in Fig. 5(a). Therefore, if we can construct a core network in which the delay among leader peers is less than 30 msecs, we can offer video conferencing with the end-to-end delay less than 100 msecs, which is smaller than the recommended one way delay for voice communication [9].

Figure 5(b) shows the average and maximum delay between the leader peer and candidates. When comparing to Fig. 5(a), the delay for candidates is less than that for all peers. It means that speakers have better and smoother conversation. We should note here that the delay for candidates remains constant after 300 seconds. This is because that the candidates have moved near the root by 50 speeches before 300 seconds.

Figures 6(a) and 6(b) illustrate how a tree was reorganized in a certain simulation run. In these figures, filled circles correspond to the candidates and open circles indicate other peers. The figures show that the tree reorganization mechanism reduces the height of the tree. With the 1000 simulation experiments, the average hop distance from the leader peer to all peers is reduced from about 7 hops to about 4 hops, and the maximum hops changes from about 14 hops to 10 hops by promotion related to fanout comparison and completion. Furthermore, we can see that the candidates have moved near the root of the tree. With 1000 simulation experiments, the average hop distance from the leader peer and candidates decreases from about 7 hops to about 3 hops, and the maximum hops changes from about 10 hops to 6 hops by promotion for speaking. However, all candidates are not necessarily located near the root depending on the timing of speaking or the duration of speaking, as shown in Fig. 6(b).

The number of messages received per second of a single peer is 0.0839 on average and 1.95 at maximum. By assuming the message size to be 5 Bytes and the packet size including the header to be 33 Bytes, the bandwidth consumed by control messages for a peer is 22 bps on average and 515 bps at maximum. This is very small compared to the rate of the data streaming in video conferencing which ranges from 64 kbps to 8 Mbps.

4 Conclusion

In this paper, we proposed a network construction method for a scalable P2P conferencing system consisting of the network construction mechanism, the tree reorganization mechanism, and the tree recovery mechanism. We evaluated the tree reorganization mechanism through the simulation experiments. We showed that the tree reorganization mechanism can offer smooth conferencing with low delay by moving peers which have high bandwidth or/and are actively speaking to the top of the distribution tree. In addition, we showed that the load of the control messages for the mechanism is very low.

However, we conducted the simulation under the assumption that no failure occurs. As one of future works, we will evaluate our method with peers dynamically joining/leaving and extend our experiments to several trees.

References

1. : (Smoothcom) available at `http://www.zetta.co.jp/ecom/smoothcom/`.
2. : (Warpvision) available at `http://www.ocn.ne.jp/business/infra/warpvision/`.
3. Jin, X., Cheng, K.L., Chan, S.H.: Sim: Scalable island multicast for peer-to-peer media streaming. In: Proceedings of IEEE International Conference on Multimedia Expo (ICME),. (2006)
4. Zhang, R., Hu, Y.C.: Borg: a hybrid protocol for scalable application-level multicast in peer-to-peer networks. In: NOSSDAV '03: Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Vi deo, New York, NY, USA, ACM Press (2003) 172–179
5. Suetsugu, S., Wakamiya, N., Murata, M.: A hybrid video streaming scheme on hierarchical P2P networks. In: Proceedings of Internet and Multimedia System and Applications 2005 (EuroIMSA 2005). (2005) 240–245
6. Barabasi, A., Albert, R.: Emergence of scaling in random networks. Science **286** (1999) 509–512
7. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: An approach to universal topology generation. MASCOTS (2001) 346
8. Kawahara, T.: Recognition and understanding of voice communication among humans (in japanese). Technical report, Kyoto University (2005) available at `http://www.ar.media.kyoto-u.ac.jp/lab/project/`.
9. ITU-T: Recommendation G.114 - one-way transmission time. Switzerland (2003)