PEER-TO-PEER MULTIPOINT VIDEO CONFERENCING USING LAYERED VIDEO

İstemi Ekin Akkuş, M. Reha Civanlar, Öznur Özkasap

Department of Computer Engineering
Koç University, Istanbul, Turkey
{iakkus|rcivanlar|oozkasap}@ku.edu.tr

ABSTRACT

A new peer-to-peer architecture for multipoint video conferencing that targets end points with low bandwidth network connections (single video in and out) is presented. It enables end points to create a multipoint conference without any additional networking and computing resources than what is needed for a point-to-point conference. The new architecture is based on layered video coding with two layers at the end points. It allows each conference participant to see any other participant at any given time under *all* multipoint configurations of any number of users, with a caveat that some participants may have to receive only the base layer video. Layered encoding techniques usable within this architecture are described. A protocol for implementation of the new approach has been developed and simulated. Its performance is analyzed.

Keywords: Multimedia communication, Distributed Computing, Teleconferencing, Internet

1. INTRODUCTION

Even though several instant messaging (IM) applications (e.g. Microsoft Messenger™) and voice over IP (VoIP) solutions (e.g. Skype™) allow pair-wise video communications in multi-user chat rooms or multipoint (MP) audio conferences, MP video is still not a popular application. This is mostly because of the fact that low bandwidth connections (e.g., ordinary modem over a phone line or wireless GPRS) that are barely enough for point-to-point video communications make more than one video connection infeasible. Moreover, users tend to consume as much of the available bandwidth as possible to increase their video quality and, hence, a MP video system that increases the demand for bandwidth can't be popular.

The bandwidth demand of a MP video system can be reduced using network based equipment called *multi point control unit* (MCU) [1]. The MCU acts as a single-point recipient for each participant, thus needing a large bandwidth connection itself. It prepares a multipoint video representation that can fit into a smaller bandwidth and sends it to each participant. Because of the complexity and cost of the operations of the MCUs, they are mostly used by large business applications that can afford such equipment.

Multicasting is another approach to reduce bandwidth demands of MP video conferencing whenever the underlying network supports it. The additional advantage of a multicasting-based solution is the reduced operational complexity [2]. Unfortunately, this approach is not applicable due to the fact that native mode multicasting on the global Internet has not been realized.

An alternative approach to MP video conferencing is presented in [3]. The system is based on the use of a distributed peer-to-peer (P2P) architecture which does not need any special hardware or network infrastructure support. There is no additional networking and computing resources needed at the end points more than that of a point-to-point video conference. In this approach; however, each participant could see one other selected participant under *most* practical cases. There are cases where some participants can't be seen by some others and the number of such cases increases with the number of participants.

In this paper, we present an extension to this architecture where we use layered video coding to make the system work under *all* cases. That is, with the new architecture each participant can see any other selected participant *anytime* for any number of participants. Two approaches are described: one using scalable video and another one using multiple descriptions. Scalable video coding techniques are gaining popularity with H264/SVC allowing encoding of the video in different quality layers so that according to the bandwidth restrictions corresponding layers can be transmitted [4]. Quality is increased by using more layers. Multiple description coding is another alternative for transmitting video [5]. In this approach, the video is encoded as two descriptions so that each of them can be used to display the video in an acceptable quality. The quality is increased if more than one description is received.

As P2P systems are becoming extremely popular, they find diverse applications. In [6], an implementation of video conferencing through an end system multicast has been reported. Although this system employed P2P techniques for MP video conferencing, it assumes that participants have large upstream bandwidths. In [7], a layered P2P streaming mechanism for on-demand media distribution is proposed. This work points out the asynchrony of user requests and heterogeneity of peer network bandwidth. As the solution, cache-and-relay and layer-encoded streaming techniques are proposed. The solution has been shown to be efficient at utilizing peers' bandwidth, scalable at saving server bandwidth consumption, and optimal at maximizing streaming quality of peers. Another P2P solution for MP video conferencing is also given in [8], using a centralized architecture. Our P2P approach for MP video conferencing, however, focuses on the needs of users with just enough bandwidth connections for single video without any single point of failure and central server.

Next section describes our P2P approach and how it employs layered video. Section 3 is about various system optimization considerations. Section 4 presents our results and discussions. We conclude in Section 5.

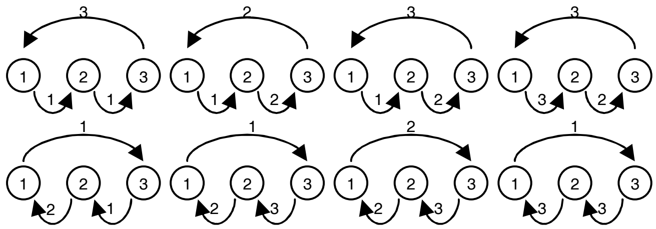


Figure 1: Configurations for a three-participant conference (Numbers on arrows indicate which video signal is used in that upstream) Each participant can see any other one.

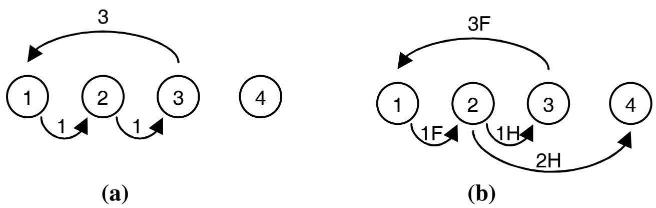


Figure 2: a) Participant 4 can not see participant 2.
b) Participant 4's request can be granted.
(F stands for *full* quality video (base & enhancement)
H stands for *half* quality video (base))

2. P2P APPROACH USING LAYERED VIDEO

A. Handling of Multipoint Configurations: In the P2P approach presented in [3], it was assumed that each participant could produce, send, and receive only one video signal at any given time. This way, the networking and computing resources of a MP video conference did not exceed the needs of a point-to-point video conference. A participant in this scheme would have to pass a video signal that it receives to others at some cases. Although this does not bring much additional burden on the peer, the entire bandwidth available for upload is used up, and the participant could not grant a request for its own video. Thus, a peer sending its own video could not act as an intermediate peer and an intermediate peer could not send its own video signal.

The cases that could emerge on a video conferencing with three participants are depicted in Figure 1. As can be seen, there is always a solution to grant every participant's every request. However, when a fourth user requests the video of an intermediate peer, like shown in Figure 2(a), the request can't be met since the upstream bandwidth is already in use.

In [3], a participant sending its own video signal is defined as a *head* of a *chain*. Chains consist of members who receive the same video signal. A *relay node* is a participant who forwards another participant's video. With our extension, a participant may be the head of a chain and a relay node (i.e. it is a member of a chain) at the same time.

Our extension employs layered video with two layers. One of the layers is the *base* and the other is the *enhancement*. The two layers have *almost* equal bandwidths, so that sending a base layer and an enhancement layer, is not different from sending two different base layers. A participant receiving a base layer and the corresponding enhancement layer is able to view the video signal in *full* quality whereas one receiving only the base layer can view *reduced* quality video. This allowed us to achieve a solution under *all* cases without having to reject any video request from any participant at anytime. Our extension and the way it solves the described problem can be seen in the example shown in Figure 2(b). The pseudo code describing the actions performed by a participant who receives a video request is given below.

```
Participant u requests video signal of v
If v is chain head
    If v is relay node
        If u can pass through base layer
            Insert u into chain of v with base only
        Else
            Add u at the end of chain of v with base only
    Else
        If u can pass through full
            Insert u into chain of v with full
        Else
            Add u at the end of chain of v with base only
Else
    If v is relay node
        Start new chain with base only and add u
    Else
        Start new chain with full and add u
```

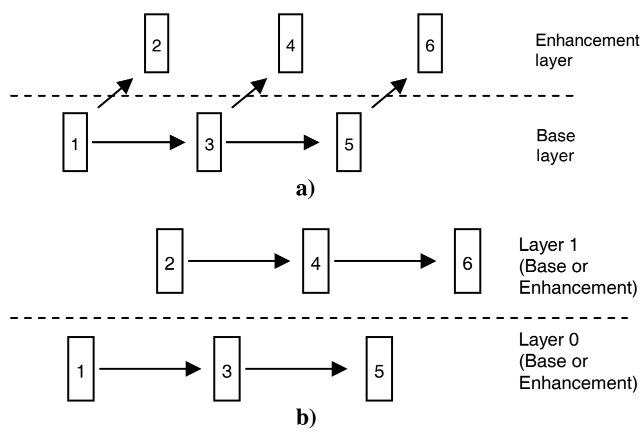


Figure 3: a) Scalable Video Approach
b) Multiple Description Approach
(Arrows indicate which frame is a reference for the next)

B. Layered Video Solutions: Figure 3 shows two possible layered video encoding arrangements usable within our architecture. In the scalable approach (a), frames are encoded simply by using the odd numbered frames before them as reference frames (e.g., 3 is reference for 4 and 5, 1 is reference for 2 and 3 (and thus, for 4 and 5 also) etc.) Such an arrangement can be implemented with minor modifications on the standard encoders [9] or scalable coders [10].

In the scalable approach, the base layer consists of the *odd* numbered frames and the enhancement layer consists of the *even* numbered frames. In the multiple description approach (b), *odd* numbered frames and even numbered frames may be predicted only from each other, creating two independently decodable threads. This approach may also be used to increase the system's loss resilience by allowing forwarding of the description experiencing smaller number of packet losses at the relay nodes. On the other hand, scalable video is more efficient in terms of bandwidth use because the enhancement layer would have better coding efficiency (e.g. frame 6 in scalable coding would be encoded by using frame 5, however, in multiple description, it is encoded by using frame 4, which is further away in time). As an example, bit rates and PSNR values of a standard sequence (Foreman), by employing both approaches using an H264/AVC codec [11] are presented in Table 1. PSNR values in the first and second rows are unusually similar, but, we should note that their framerates are different.

	Y	U	V	Bit rate (kb/s)
Base+Enh	31.4	37.6	38.3	29.2 + 19.4 = 48.6
Only Base	31.5	37.7	38.3	29.2
No Layers	32.3	37.9	38.6	51.09
MD	31.5	37.6	38.3	29.2 + 29.3 = 58.5

Table 1: Average PSNR values and bit rates

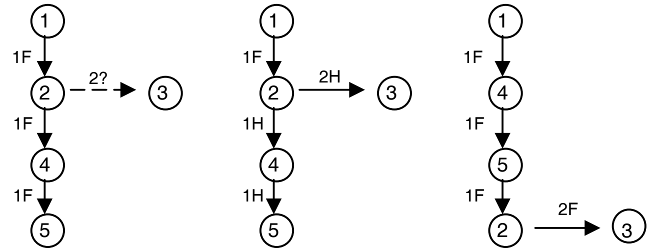
3. OPTIMIZATIONS

Chain configuration optimizations can be performed in order to maximize the number of participants that receive *full* quality video in a particular configuration. In other words, let *n* be the number of participants and *r* be the set of *video requests,* defining a particular configuration. The number of *full* quality receivers is $k = f(n, r)$. The aim of the optimizations is to *maximize k,* when the number of participants changes and/or the request of a participant changes, namely $k' = f(n', r')$.

For example, assume that a participant, e.g. 2, is relaying another participant's, e.g. 1's, video signal at full quality, that is, base plus enhancement layers. When yet another participant, e.g. 3, requests 2's own video signal, 2 has to drop the relayed video signal (of 1) to base layer only, so that it can send its own video to 3. This makes all participants receiving the relayed video from 2, and participant 3 to receive *half* quality video. Assuming that 2 is located right after 1 in a long chain, letting it relay *half* quality video would reduce the received video quality for a large number of participants. However, if 2 could be moved to the end of the chain, this large number of participants can continue to receive *full* quality video. Figure 4 shows this situation.

If in the configuration in Figure 4, participant 5 was also sending its own video signal, then moving 2 to the end of the chain would cause all the participants in 5's chain to receive *half* quality video. In this case, chain lengths for 2 and 5 can be compared and the participant with a longer chain can be moved to the end. For instance, if 5 has a chain length of two, then moving participant 2 to the end causes a total of three participants to receive *half* quality video (the chain members of 5, i.e. two participants, plus participant 2). On the other hand, moving participant 2 just *before the end* would cause only two participants to receive *half* quality video (the participant that requested video from 2, i.e. 3, and participant 5).

Similarly, when inserting participants into chains, the lengths of the involved chains should be taken into consideration. If a participant is the head of a chain sending *full* quality video, we should avoid using it as a relay node. As an example, assume that a participant, P, that is a chain head sending *full* quality video, requests participant T's video signal and T is already sending at *half* quality. In this case, P could drop its video to *half* and relay



(a) **(b)** **(c)**

Figure 4: An example of chain optimization
a) Participant 3 requests a relaying participant's video
b) Chain without optimization
c) Chain after Participant 2 is moved to end

T's video signal. However, doing so would cause the chain members of P receive *half* quality video. A better solution would be adding P to the end of the chain, so that it would continue to send its video at full quality. If the last member, L, of the chain is also a chain head sending its video; however, a comparison between P's and L's chain lengths should be carried out. The participant with longer chain would go to the end of the chain, minimizing the number of *half* quality video receivers. This greedy approach ensures that every time a participant requests video from another one, the configuration stays with maximum number of *full* quality receivers.

While making these optimizations, however, some configuration messages need to be sent, because checking chain lengths or mobility of members in the chain require exchanging information between chain head and members. Although these can be done in one message, there may still be delay before a configuration is updated. Therefore, some optimizations may be skipped whenever low delay is more important than quality. The optimizations can be performed after the requested video is provided immediately using a suboptimal configuration. Besides this issue, the geographic location of the users would also play a role on the ordering of the participants in a chain. In that case, the trade-off between maximizing the number of *full* quality receivers and minimizing the maximum delay any participant experiences still remains and can be left to the users' choice.

Messages used in the P2P system are listed in [3]. In the new architecture, the extensibility messages in this list become obsolete since every chain becomes extensible (by using *half*). Instead, for optimization purposes, two other messages are added:

(i) *Chain length check message:* Sent by the head to discover whether the last member or the requester has longer chain.

(ii) *Chain length info message:* Sent by a participant (member of a chain or requester) to inform the head of the chain length and status of the chain (*full* or *half*).

4. SIMULATION RESULTS & DISCUSSIONS

In the simulations, we generated *all* possible cases for a given number of participants where each participant requests a video signal. Using the algorithm given in Section 2 in pseudo code and employing optimizations described in Section 3, the chains were obtained and analyzed. Figure 5 shows that with the increasing number of participants, the probability that a participant receives half quality video increases. In Figure 6, the percentage of the total number of half quality receivers in

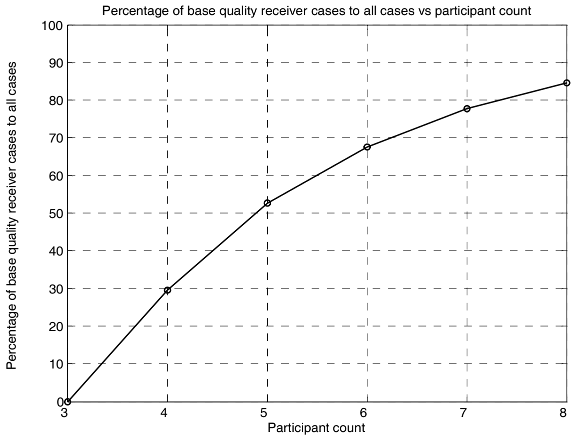


Figure 5: Percentage of configurations containing half quality receivers versus number of participants.

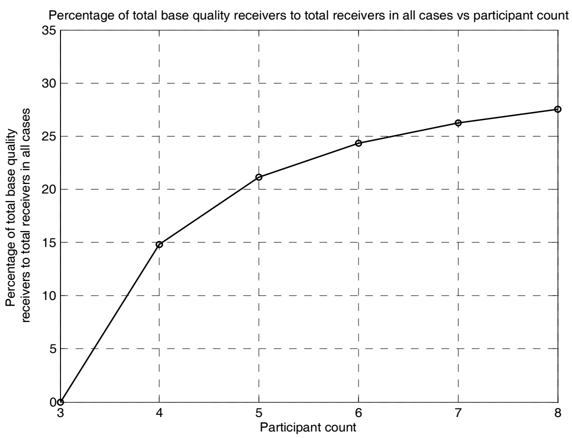


Figure 6: Percentage of half quality receivers to all receivers under all configurations versus number of participants

all receivers, under all possible cases is shown. Increasing the number of participants would increase the number of possible cases and thus, the total number of half quality receivers as well. However, this increase is asymptotic and the ratio of the average number of half quality receivers to the number of participants decays. This is shown in Figure 7.

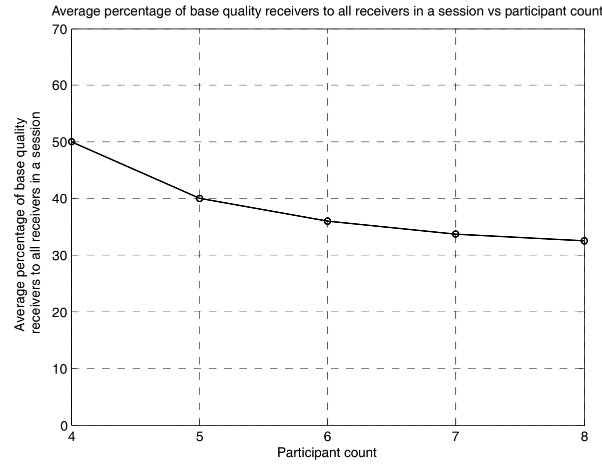


Figure 7: Average percentage of half quality receivers versus number of participants.

5. CONCLUSION

We presented an extension to the P2P architecture for MP videoconferencing [3]. Employing layered video, this extension makes it possible to find a feasible solution under *all* configurations, so that a participant's video request can always be granted at anytime. Simulations show that with the increasing number of participants, the use of layers and thus, *half* quality video receivers is inevitable; however, the ratio of the half quality receivers to the total number of participants remains under 50%. The presented architecture is very easy to implement using existing encoding and networking infrastructures.

6. REFERENCES

[1] ITU-T Study Group XV - Recommendation H.231, "Multipoint Control Units for audiovisual systems using digital channels up to 1920 kbits/s", March 1993.

[2] M.R. Civanlar, R. D. Gaglianello, G. L. Cash, "Efficient Multi-Resolution, Multi-Stream Video Systems Using Standard Codecs, Journal of VLSI Signal Processing, 1997.

[3] M. Reha Civanlar, Oznur Ozkasap, Tahir Celebi, "Peer-to-peer multipoint videoconferencing on the Internet", *Signal Processing*: Image Communication 20 (2005) 743-754

[4] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), May 2003.

[5] Vivek K. Goyal, "Multiple Description Coding: Compression Meets the Network", IEEE Signal Processing Magazine, Volume 18, Issue 5, Sept. 2001 Page(s):74 – 93

[6] Yang-hua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture", Proceedings of ACM SIGCOMM'01, San Diego, California, August 2001.

[7] Y. Cui, K. Nahrstedt, "Layered Peer-to-Peer Streaming", Proceedings of NOSSDAV'03, June 2003, Monterey, California.

[8] M. Hosseini, N. D. Georganas, "Design of a Multi-Sender 3D Videoconferencing Application over an End System Multicast Protocol", ACM, Multimedia 2003, Berkeley, CA, Nov 2003.

[9] JM Software Available at http://ftp3.itu.ch/av-arch/jvt-site/reference_software/

[10] JSVM Software Available at CVS Repository of JSVM :pserver:jvtuser@garcon.ient.rwth-aachen.de:/cvs/jvt

[11] x264 Software Available at http://www.videolan.org/x264.html