The Flow Control of Audio Data Using Distributed Terminal Mixing in Multi-point Communication

Young-Mi Kim and Dae-Joon Hwang

School of Information and Communication Engineering, Sungkyunkwan University, 300 Chunchun-Dong, Jangan-Ku, Suwon-city, South-Korea, 440-746
ultraym@skku.edu, djhwang@skku.ac.kr

Abstract. This paper describes an efficient audio flow control method in the point of quantitative performance using audio-mixing, compared to existing P2P(Peer To Peer) method. In comparison with existing P2P method, using central mixing and distributed terminal mixing method, we achieved advance at the point of global network usage and each terminal's CPU load, and additionally we expect more session, more terminal can be served by same amount of network bandwidth and computers. By using P2P method in audio communication, speaker and listener must connect to each other. So it has the critical defect that as the participants grows more and more, the network bandwidth usage, each terminal's CPU load will grows rapidly. So the number of participants in same session will be extremely restricted. In comparison with P2P method, the central mixing method has the great advantage at the points of network usage and terminals CPU load. Regardless of the number of speakers and listeners, all the participants can speak and listen with all other participants by using just one stream's amount of data size and CPU load. But all the network usages and CPU loads of "Audio decompression->Buffering->Mixing->Audio Compression" are concentrated on central server. So the number of sessions and terminals can be participated in one server will be highly restricted. This study solves the problems of server's CPU load and network load by using the distributed terminal mixing method.

1 Introduction

As Internet technologies have dramatically improved and Internet services have been widely expanded and adopted in recent years, the demand for multimedia data services is much higher than ever before in our daily lives. This is remarkably true with real-time audio/video communication services, which have been deployed very much in quantity and require ensuring accurate and real-time transfer based on data loss restoration, however, considerable data volume led to too much load on networks.

In addition, the rapid advancement of computer and communication network technologies contributed to make the speed of processor as well as network much faster. However, in the other hand, there appeared more Internet-users, protocols and programs requiring considerable throughput. So there has always been and will continue to be some kind of effort to minimize required computer processing capacity and network usage in order to save costs.

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 671 – 680, 2006.

As we do not have any specific organization responsible for controlling network resources, each network program has its own way of leveraging allocated bandwidth and servicing users. Even if some two programs' functionalities are identical, their performance and/or network bandwidth consumption may be different from each other, depending on transfer method. Especially in multi-point multimedia network programs, there are a lot of contributing factors to this kind of difference. These factors can be grouped into multimedia data compression and multimedia data transfer. Multimedia data compression has been evolved consistently. Standard organizations continue to provide CODECs with better video output and compression rate, which are being deployed to a wide variety of areas, including multimedia-enabled programs, consumer electronics, etc. The development of multimedia transfer technology which helps save network resources doesn't seem so fast as that of CODEC, however. The reason is in most cases, P2P-based approach has been deployed to multi-point area, which doesn't make full use of the opportunities for saving bandwidth. This aspect would lead to a condition where there are only several multi-point communication sessions occupying the entire network.

There is a protocol for real-time multi-point stream, called IP-multicasting.[1][6] With that protocol, we can deliver the stream to desired targets using just one transmission, but because there are not enough routers supporting IP-multicasting, in practice, we must use also another protocol with it to complete all transmission. And IP-multicasting has no gain on CPU load. And one of important defect is that it can't guarantee the delivery of data we sent. It's very difficult work to restore the lost packets to get a continuous stream in real-time in especially multi-point communication.[2]

In addition, CODEC evolvement required more CPU load. In a multi-point multimedia communication, you should transfer data to and/or receive & process data from more than one person. To process more than one stream in a single computer, it should be assumed that the computer has enough processing capacity. We cannot always make sure any user's computer would meet the requirement, however.[3][4]

Although it's not the main topic of this study, we should consider the sequence from the capturing audio signal to playing audio "analog audio signal->conversion to digital signal -> compression -> sending network packets -> receiving network packets -> decompression -> conversion from digital to analog signal -> playing with audio output device". And the compression and decompression occupy the most of load on CPU.

In this paper, we're going to focus on improvement on these issues below.

- Global network usage
- Server / terminal network usage
- Server / terminal CPU usage
- The number of sessions and terminals that can be accommodated on the same number of server and network bandwidth.

We suggested the central mixing method for reduction of global network usage and terminal's CPU load. This is the method that all the audio data must be collected in central mixing server first, and the mixing server mixes the data and sends the mixed data to each participant terminal.

But the central mixing method has the critical defect that network loads and CPU loads are centralized on server. This brought that the server can't accommodate as much sessions and terminals as the P2P method.

With supplement the central mixing method, we suggest the distributed terminal mixing method that maximizes the usage of terminals' network and CPU resources.

2 Reference Studies

Looking at the latest audio communication technologies and their more traditional counterparts, the most of multi-point audio communication use the P2P method in audio communication. Each participant sends their audio data to each other participants and each participant receives audio data from each other participants.[2][5][7] For the network data packet standardization, we uses the RTP(Real-time Transport Protocol) in packet type. [9]

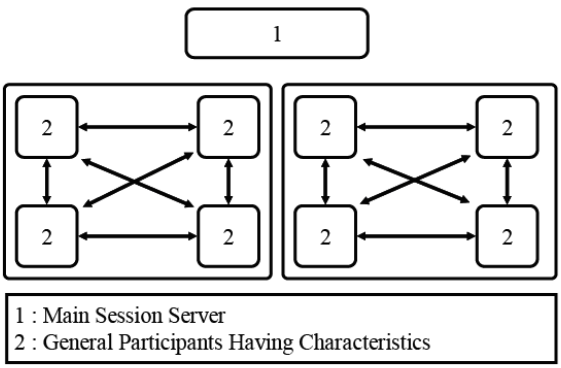

Fig. 1. P2P-based multi-point audio communication architecture

The existing P2P typed multi-point audio communication had the volume of network data increased as much as N*(N-1)*2(where 'N' is the number of participants, 2 is for listening and speaking), just as in Fig. 1. In order to listen to all speakers, the listener had to receive (N-1) data.

Not the only audio data size, the CPU load in decompressing the compressed audio data should be also considered. So with more participants, the communication data volume and CPU load from data decompression would be increased rapidly, which in turn, became one of the main reasons for placing limit on the number of simultaneous speech and its audience. Actually, in an ADSL (Asymmetric Digital Subscriber Line)-based environment, only about 4 people could speak and listen concurrently. The following figure shows the case that a session consists of 4 participants, each of whom should perform one compression and three transfers/ receives/decompressions in order to send audio data to and get the data from others. As more people participate, each terminal will experience rapid increase of network/CPU load.

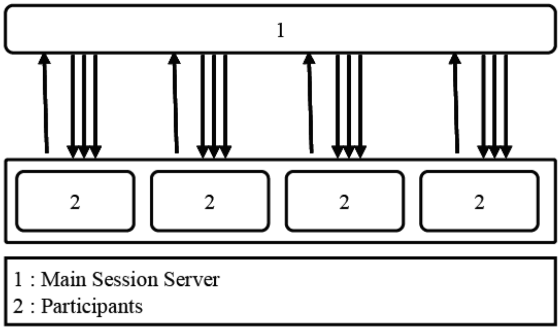

Fig. 2. Centralized relay-based multi-point audio communication architecture

In response to this problem, centralized relay (at service provider's equipments or servers) such as Fig. 2 was suggested.

In a centralized relay approach in Fig. 2, each user transfers his/her own data to the server just once, which would relay the data to other users. Whereas transfer data to the server will be done just once independently of participants, receiving and decompression should be done (N-1) times just like in P2P-based. This solution will, therefore, reduce only data size from terminals to server as compared to P2P. It is useful in such a network environment that receiving-bandwidth is much larger than the sending-bandwidth (ex: ADSL).

3 The Structure of Algorithm

In this study we aimed at reduction of network usage and terminals' CPU load, using central mixing method compared with P2P method. And we aimed at distribution of server's network and CPU load, for the same amount of servers to accommodate as more sessions and terminals as possible.

In this study, the base of idea is the audio's key characteristics that many audio streams can be mixed into the stream equivalent to just one stream in size. In real life, when various sounds are audible, we can just hear some loudest sounds. This is

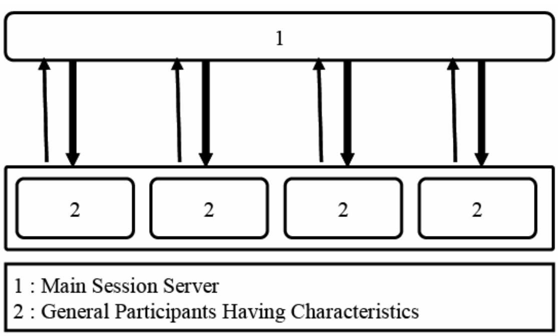

Fig. 3. Centralized mixing-based multi-point audio communication architecture

because the sounds having different frequency counterbalance each other. With other media such as video, text, we can't see these mixing effects.

In a centralized audio mixing approach in Fig. 3, along with outgoing network data size at terminals shown in Fig. 2, incoming network data size and CPU load during decompression can be reduced. Each terminal compresses its own audio data and sends it to the server, which in turn, decompresses all data and mixes them. The size of mixed data will be of only one audio stream. Mixed data will be compressed and distributed to each terminal, which in turn, receives just one stream, decompresses it and sends the decompressed data to audio output, in order to listen to all the other participants' audio.

Irrespective of the number of participants, by sending and receiving just one stream size data, all the participants can speak to all the participants and can listen to all the other participants' audio. This is the remarkable advantage compared with P2P or central relay method.

However, with this approach, when many session being is created (a session means a communication group), the main server had to get lots of network load and processing load caused by decompression, mixing and compression. Because of this, the cost of load balancing and network maintenance will rise very rapidly, according to the number of sessions and terminals. So it didn't seem to be so feasible for multi-user configuration. As CPU load from audio compression/decompression is considerably heavy, server load balancing cost may be much higher than the network cost.

In this paper, we suggest the distributed terminal mixing method to minimize processing time and network loads on servers in central mixing method, and to minimizing processing time and network loads on terminals in P2P method.

In this paper, we're providing a scenario where each session has its own mixing agent responsible for processing and transferring the session's audio data, and there is an audio data relay server which helps network transfer in case network capacity is not enough.

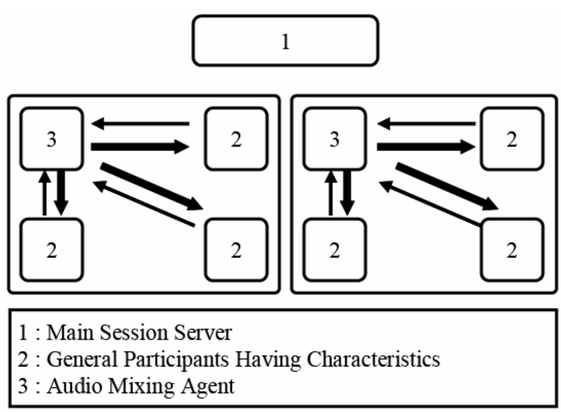

Fig. 4. Distributed terminal mixing audio communication architecture

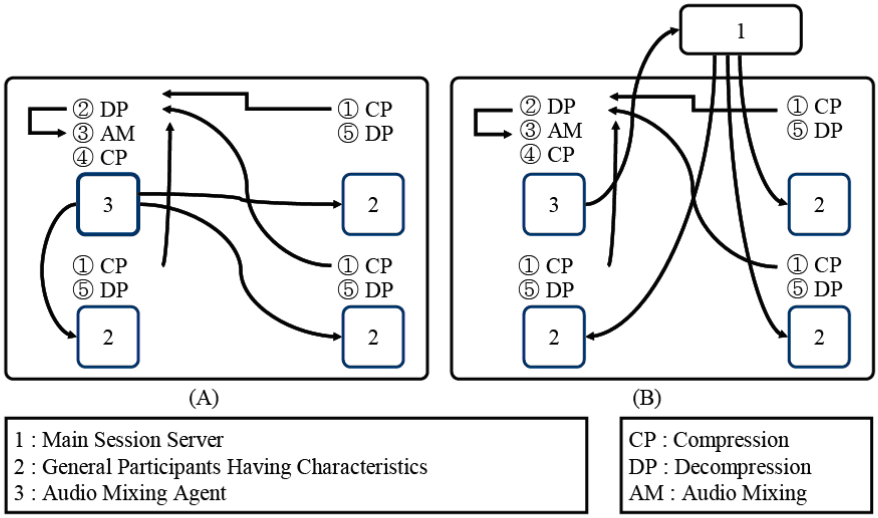

Fig. 5. Two detailed algorithms for the distributed terminal mixing method

The advanced architecture provided here is shown in Fig. 4. (A) and (B) in Fig. 5 represent the two data processing / transfer algorithms for implementing Fig. 4.

In (A) in Fig. 5, a mixing agent terminal, one of the participants takes the role of the main server in centralized mixing method. The mixing server does nothing on the audio data. The normal participants (counterpart of the mixing agent terminal) send their compressed audio data to the mixing agent terminal. And the mixing agent terminal receives all other participants' stream and stores them into each stream buffer. And then, it extracts some audio samples in each time slot, and mixes it into one stream. And the mixing agent terminal sends the mixed stream to all other participants. Other participants receive the stream, and decompress it and output to the audio device. Normal ADSL has a wide download network bandwidth relative to upload network bandwidth. So (B) in Fig. 5 that the relay server relieves the network transmission is recommended rather than (A) in Fig. 5 in these network environment.

In this method, the relay server would transmit the audio data that is mixed and compressed by the mixing agent terminal. In comparison with Fig. 3, the central mixing method, there is no load on the central server for the compression and mixing and decompression. There is advantage over the central mixing method in that only one stream can be enough to all participants' speaking and listening instead of all the participants' stream. In case of large amount of stream data that one relay server can't afford, more relay servers can be used to distribute network loads.

In the Fig. 5, we are going to call (A) the pure mixing agent method, and (B) the mixing agent method having relay server.

We can reconfigure the network topology according to the various situations and conditions by using the hybrid method of (A) and (B) in Fig. 5. Let's take a look into some useful topology example.

The first example is this. The mixing agent terminal processes all the participants' stream until its capability of CPU and network come to a limit. After the limit, central mixing-relay server would process the next participants joining same session. This is reasonably nice for distribution of CPU and network load besides the simplicity in implementation.

So the agent should have enough CPU and network capacity to process all the session data on each terminal.

The mixing agent from (B) in Fig. 5 collects data from participants, mixes and compresses the data, transferring it to the relay server.

If you want for the servers not to be used for mixing or relaying, you can distribute CPU and network load by constructing the tree transfer topology. When the mixing agent terminal comes to a limit in CPU's processing power or network bandwidth, it selects another mixing agent terminal among its children terminal, and then makes it process the some of terminals. The children mixing agent terminal sends its mixed data to there parent mixing agent terminal. In sending audio data mixed, the same topology can be used. For the better quality of distribution, different topology between the upstream and downstream is recommended. This type of transfer topology has a week point in reconstruction of tree when one of mixing agent terminals leaves the session. A good reconstructing algorithm must be accompanied not to degrade audio stream quality. But this is a good adaptation at the point of view of saving maintenance cost.

By adjusting the (A) and (B) in Fig. 5, we can make many useful topologies according to the environment.

The (A) algorithm in Fig. 5 is used In the data for comparison in [Table 1].

Table 1. The numeric comparison between flow control method

- In a N-concurrent user configuration (all can speak and listen)
- The Network Usage Unit: Amount of one stream
- CPU load type: Compression, Decompression, Audio Mixing
- S: Sending , R:Receiving, C:Compression, D:Decompression

Criteria	**P2P Method**	Central Relay Method	**Central Mixing Method**	**Distributed Terminal Mixing Method**	
				Ordinary Participant	**Mixing Agent**
Network Usage Per Participant	S(N-1) R(N-1)	S(1) R(N-1)	S(1) R(1)	S(1) R(1)	S(N-1) R(N-1)
Network Usage In Server	0	S(N*(N-1)) R(N)	S(N) R(N)	0	
The number of Comp. /Decompression Per Participant (CPU load)	C(1) D(N-1)	C(1) D(N-1)	C(1) D(1)	C(1) D(1)	C(1) D(N-1)
The number of Comp./Decomp. In Server(CPU load)	0	0	C(1) D(N)	0	
The number of Audio Mixing Per Participant (CPU load)	1	1	0	0	1
The number of Audio Mixing In Server (CPU load)	0	0	1	0	
Global Network Usage	N(N-1)	N*N	N*2	(N-1)*2	

Seeing the above table, the global network usage in the central mixing method and the distributed terminal mixing method is much less than the P2P method as the N is growing. At the point of view of the global network usage, the distributed terminal mixing method has same amount of network usage, N*2 to central mixing method. But at the point of view of server, because the distributed terminal mixing method does not use server's network, it is far better at the point of view of maintenance cost than the central mixing method.

Taking the account of the CPU load, in P2P method, all participant terminals compresses one time, and decompress N-1 times. In central mixing method, all participant terminals compress and decompress just one time to speak to all participants and to listen to all participants. In distributed terminal mixing method, the ordinary participant terminals compress and decompress once just like central mixing method except that the mixing agent terminal decompresses N-1 times and compress once. Although the central mixing method is better than the distributed terminal mixing method in this respect, the latter is far better than the former in respect to CPU load on server.

The comparison of network usage data gotten from simulation is shown in Fig. 6

All nodes can listen and speak simultaneously. We used 44.1 khz, 16 bits, stereo samples, bitrate of 128 kbps in the simulation. In network, TCP is used. As it shows in Fig. 6, as the number of nodes increase, the distributed missing method is far better than others.

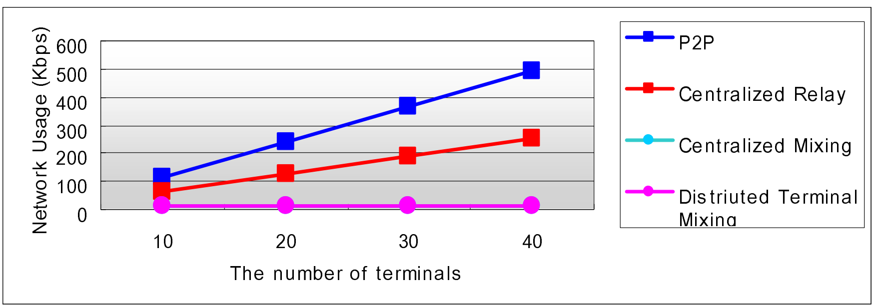

Fig. 6. The network usage for one normal participant

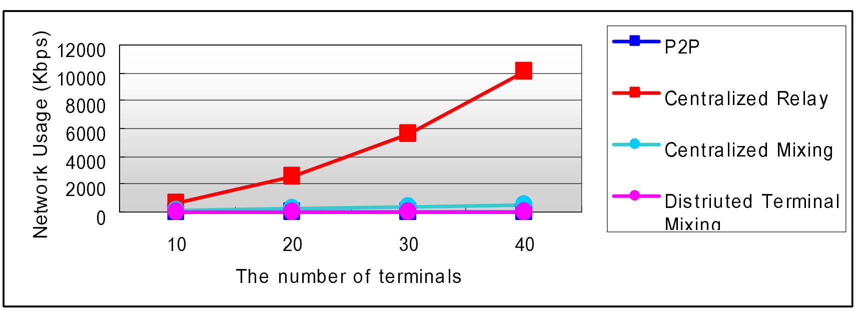

Fig. 7. The network usage in server

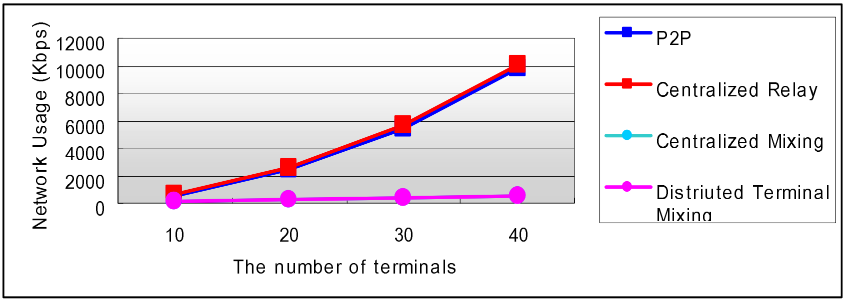

Fig. 8. The global network usage

Fig. 6 shows the network usage for one normal participant according to the number of terminals. In this figure, the centralized mixing method and the distributed terminal mixing method have the same values. And they have always the same value of 2 stream's bandwidth, regardless of the number of terminals.

Fig. 7 shows the network usage in server according to the number of terminals. In this figure, the P2P method and the distributed terminal mixing method have the value of 0 in all cases. This is because in these methods, there is no intervention of server to the communication.

Fig.8 shows the global network usage value, the sum of all terminals' network usage and server's network usage. In this figure, we could see that the network usage increases by geometric progression by the number of terminals, in P2P method and the centralized relay method. But in the centralized mixing method and distributed terminal mixing method, the network usage increases in proportion to the number of terminals. At the point of view of only global network usage, the distributed terminal mixing method is the best method.

In this study, we use some method to select a mixing agent. The first case is that the president takes the responsibility of the mixing agent. In that case there is assumption that the terminal CPU and network is good enough to process the job. And the next case is that we select the mixing agent by measuring and comparing the CPU and network resources of all terminals. In this case, too many changes of topology by using dynamic selection of the mixing agent terminal will result to degradation of audio stream quality. It's recommended not to change the topology too frequently.

4 Conclusion and Further Studies

In this paper, in order to solve the problem of P2P based multi-point audio communication we suggested two major methods. The one is the central mixing method that uses audio's special characteristic that mixing multiple streams makes only one stream. And the other is the distributed terminal mixing method that makes some of the terminals share the CPU and network load of server. We suggested also some adaptations of the two methods. In addition to solve the problem of the existing P2P method, this is far advanced audio flow control method in respect to minimizing maintenance cost.

The criteria for evaluating the multi-point multimedia network sessions are mainly low network usage, low CPU processing load, low maintenance cost, and high data quality.

When we apply this study in the real environment, many problems can be found according to the each terminal's CPU power, uploading capacity, downloading capacity. This study has many assumptions. For example, all terminals must have enough CPU and network capability at least one upstream and downstream. And the mixing agent terminal must be able to process all the other terminals stream data. But in real environment, there are so many possibilities that cannot meet the assumption.

Other study must be accompanied that is able to cope with these situations. For example, network topology must be changed dynamically according to the all situations without user's intervening. And it must be done without the degradation of the stream quality. Although the topology must not be so frequently changed, in case of the blocking of stream it must be changed rapidly. This study doesn't mention the methodology about dynamic topology. This touched just a possibility that dynamic topology can be constructed according to the server and the terminals' CPU and network capabilities.

References

[1] SE. Dering, "Host Extensions for IP Mlticasting", *RFC 1112, Stanford University* , Aug. 1989

[2] A.Oram, "Peer-To-Peer", *O'Reilly*, Mar. 2001

[3] S.G. Lee, "The Study On RMTL(Reliable Multipoint Transport Layer) in IP-multicasting environment", *SungKyunKwan Uni*,, Oct. 1997

[4] Y.S. So, C.Y.Choi, "The Multi-source media streaming capable of distribution the contents in P2P network", *Korea Information Communication Society,* 2004

[5] Tsutomu Kawai, Joutarou Akiyama, Minoru Okada, "Point-to-Multipoint Communication Protocol PTMP and Evaluation of Its Performance", Jan. 1998

[6] H.R. Kim, K.S. Song, J.S. Jeon, "Implementation of Multicast protocl and management of resource for the multi-point multimedia", *Korea Industrial Engineering So-ciety,* 1998

[7] W. Simpson, "The Point-to-Point Protocol (PPP) for the Transmission of Multi-protocol Datagrams over Point-to-Point Links", *RFC 1331*, May. 1992.

[8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: "A Transport Protocol for Real-Time Applications". RFC 1889, *Audio-Video Transport Working Group*. Jan. 1996.

[9] Endeavors Tech., "Introducing Peer-To-Peer", *White Paper, Endeavors Technology Inc.*, 2002

[10] Y. Kim and Y. Eom, "An Efficient Peer Connection Scheme for Pure P2P Network Environment", *Korea Information Science Society*, Feb. 2004.