Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

Design of Multi-Singing Karaoke System and its Application in Customer Finance-aided Service on Internet

Jian-Hong Wang[1, 2] Shih-Chuan Feng[1] J.-Y. Pan[1]
[1] *Department of Communication Engineering, National Chung Cheng University*
[2] *Department of Information Engineering, Chung Chou Institute of Technology*
wanghong@dragon.ccut.edu.tw

***Abstract*—Financial engineering nowadays is heavily depending on the real-time investment decision-support system while providing the suggestions to investors for rebalancing their portfolios. Because of the rapid growth of interactive technologies in internet, on-line customer finance-aided service (CFAS) is playing an important role to help discussing, exchanging valuable information or enjoying entertainment each other simultaneously based on the platform of multi-singing karaoke system. Today's online singing software allows only one-microphone performance. Duo singers can only take turns and use the same microphone. Also, they cannot hear each other simultaneously. This is one big disadvantage of online multi-singing software, hence brings the problems to the quality of CFAS. Therefore, the main purpose of this study is to create a system that allows several singers to sing in different places simultaneously and hear the other singers' voices at the same time. This system catches and analyzes the singers' signals from remote computer sources through Media Server. Also by network broadcasting, this system transmits the signals from the inviter's end to the invitee's end, which is deeply improving the quality of CFAS.**

Keywords: Online karaoke software, Media Server, internet broadcasting, financial engineering, customer finance-aided service.

I. INTRODUCTION

Online karaoke software contributes a new try at the customer relationship service between the investment broker and financial customers via multi-client collaboration mechanisms. The users on both sides of CFAS could obtain the benefits of enhancing the communication capabilities, furthering the interests of online service and delighting in entertainment in the leisure time through the bid/ask process. However, current online karaoke software provides only solo-singing function. If the meeting room is made into a karaoke room, many people will be able to sing together but still not able to sing duo. The time delay in instant interaction application that is bearable to humans is within 100 milliseconds. Therefore, the basic requirement for a duo or multiple-singing system is that the singer must start singing 100 milliseconds within reading the subtitle on the screen.

The key point of this research is to establish a system that allows people in different places to feel the presence of the others and the effect of being on at the scene. The interaction in singing shortens the distance of online pals and creates meeting opportunities. It also brings the users closer for chatting or common interests.

The structure of this paper is as follows:

Section 1 – Introduction: online karaoke software in the applications of CFAS, discussion of delay time in instant interaction, research motives and purposes

Section 2 – Literature review: including introduction of sensitivity of human ears to time delay, various time delay in voice transmission

Section 3 – System design: including system architecture, three methods of voice transmission with time delay within 100 milliseconds

Section 4 – Results: testing results and effects of the three methods, comparison of systems

Section 5 – Conclusion and discussion: including the functions achieved by this system and future development.

II. LITERATURE REVIEW

Haas Effect, also called the Precedence Effect, describes that time delay is required for humans to identify sound sources. If two sounds arrive within a very short time, humans may not be able to localize the subsequent arrival.

A. The phenomena of Hass Effect

The phenomena of Hass Effect include:

1) Humans localize the sound sources based upon the first arriving sound. If the subsequent sound arrives within 30 milliseconds, our ears do not detect it.

2) If the subsequent sound arrives within 30-50 milliseconds, our ears can localize the two sounds.

3) When the subsequent sound arrives more than 50 milliseconds, humans can identify the two sound sources. Two distinct sounds are heard.

The time delay in instant interaction that humans can bear is within 100 milliseconds. When sounds arrive more than 100 milliseconds, the overlapping of sound occurs. Hence, for simultaneous singing, the time delay must be within 100 milliseconds.

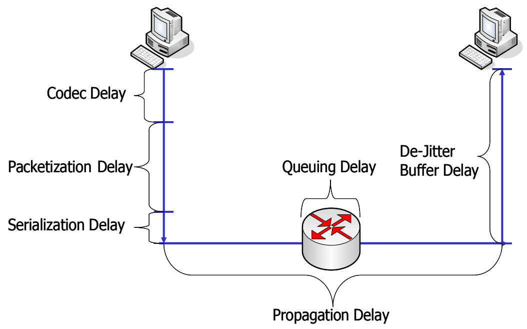


Fig. 1. The possible time delay in voice transmission.

B. *Variables of Time Delay in Voice Transmission*

Fig. 1 shows the possible time delay in voice transmission. Variables of time delay in voice transmission[3] ：

1) Codec delay is the time taken by CPU to code and compress a block of voice information. The delay depends on code complexity and processor speed.

2) Packetization delay is the time taken by the sender to accumulate voice samples. The delay is usually 20 or 30 milliseconds.

3) Queuing delay is the time that a packet is waiting to be transmitted from the buffer. The delay depends on network traffic. When the traffic is heavy, the packet will be put into the buffer and wait to be transmitted. When the traffic is light, the packet can be processed and transmitted instantly. The heavier the traffic, the longer the delay is.

4) Serialization delay is the time required to clock a dataframe onto the data interface. It is inversely propotional to the network transmission speed. For instance, in a 64kb network, the delay is 125 milliseconds for a one-byte dataframe while it is only 0.05 milliseconds for OC-3/STM-1 network cable.

5) Propagation Delay is the time taken to transmit a packet in network cable. The delay depends on the distance of the transmission media. The transmission speed of the packet signal is about 2/3 of the light speed. The equation is as follows:

Transmission time (s) = transmission distance (km) / (300000 km * 0.6)

6) De-Jitter buffer delay is the time taken to reduce the influence of the delays created by buffers.

The variable delays above do not include the time delay caused by the operation system. When executed in Windows, the delay will include the timer inaccuracy and scheduling which is within 150 milliseconds in total. The sample is taken every 20 milliseconds in packetization delay and dependent on the accuracy of timer. When an accurate timer is used, the space of CPU will be taken up and the info in buffer cannot be processed instantly. It may also cause the default. When a less accurate timer is used, the discrepancy will become larger.

III. SYSTEM DESIGN

This system allows singers to hear others instantly. If using a mixing-voice server, the sound transmission will be delayed and synchronization will not be achieved. This system does not use mixing server. However, the inviting singer can send mixed sound to Media Server. The sound then can be transmitted through Web Server in the receiving end. The Web-based transmission is user-friendly that permits users to set up personal data or browse information in an easier way. Fig. 2 shows multiple-singing illustration; Fig. 3 shows system architecture.

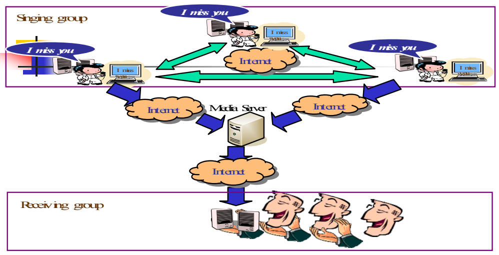


Fig. 2. Multiple-singing illustration

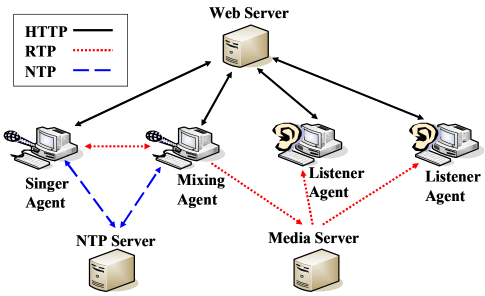


Fig. 3. System architecture

Listed below are three methods used in multiple singing karaoke system with time delay within 100 milliseconds between the play of the song by karaoke and the singing of humans. First, adjust the Windows to be the top priority and let the methods to be executed first. Fig. 4 shows time-setting of self-configuration

A. *Definition of Parameters*

1) T_d is Time delay in synchronized play of karaoke in two computers.

2) T_{os} is Time taken for the program to execute the command and the network card to transmit it. For example, Tox (X) is the time taken from the command execution by the program to the transmission of command to computer X.

3) T_n is the time taken to transmit the order on the web. For example, Tn (X) refers to the time taken to transmit the command to computer X through the internet.

4) T_p is the time taken from the network card receiving the command to the execution of the play of karaoke. For example, Tp(X) is the time that computer X takes from receiving the command to executing it.

5) T_{sync} is the time lag between the singer and NTP Server. For instance, Tsync (X) means the time lag between computer X and NTP Server.

6) T_t is the time lag in timed play of karaoke. Tt (X) means the time lag of computer X in time-setting of song-play.

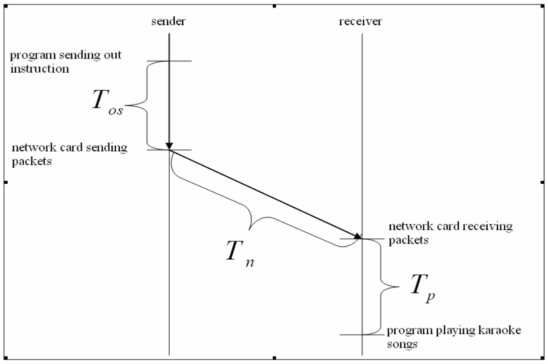

sender
receiver
program sending out instruction
T_{os}
network card sending packets
T_n
network card receiving packets
T_p
program playing karaoke songs

Fig. 4. Time-setting of self-configuration

B. Procedure of 1st method

1) The inviter sends play command to the invitee.

2) The inviter plays the song right after the transmission of command.

3) The invitee plays the song right after receiving the command.

In this situation, the time delay in synchronized play on karaokes is as follows:

$$T_d = T_{os}(Y) + T_n(Y) + T_p(Y) \qquad (2)$$

C. Procedure of 2nd method

1) A server sends song commands to all singers.

2) All singers turn on karaoke right after receiving commands. The time delay in sending commands by Server is as follows:

$$\begin{aligned} &T_d \\ &= |T_{os}(X) - T_{os}(Y)| \\ &+ |T_n(X) - T_n(Y)| \\ &+ |T_p(X) - T_p(Y)| \end{aligned} \qquad (3)$$

D. Procedure of 3rd method

1) Inviter synchronizes with NTP Server.

2) Inviter sends the time-setting for the song-play to the invitee (current time of the inviter + time postponed).

3) Invitee synchronizes with NTP Server after receiving command from inviter.

4) Invitee plays the song at the time set by inviter.

In this case, the time delay between the inviter and the invitee on synchronized play is as follows.

$$\begin{aligned} &T_d \\ &= |T_{sync}(X) - T_{sync}(Y)| \\ &+ |T_t(X) - T_t(Y)| \end{aligned} \qquad (4)$$

When there are several works with priority waiting to be executed in transmitting synchronized play commands, the time taken to obtain CPU availability will depend on the execution time of first works in queue.

Table I represents the list of sequence and the clock interruption interval is 10 milliseconds. The higher the priority, the smaller the value is. If P3 represents a program of command transmission, it has to wait 19 milliseconds to use CPU. Therefore, the heavier the traffic, the longer the hold is. Fig. 5 shows time table of CPU.

TABLE I
CURRENT SCHEDULED INFO

Schedule	Time of Execution (ms)	Priority
P1	30	1
P2	9	1
P3	17	1
P4	6	2

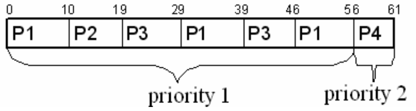
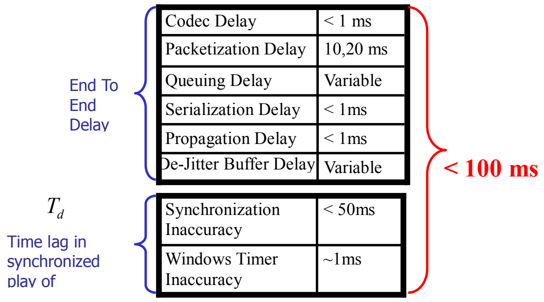


Fig. 5. Time table of CPU

In addition, it is also hard to set a timeframe due to the unpredictable time delay in internet transmission. For instance, it would take 5 milliseconds for Server to send commands to user X while it would take 65 milliseconds to send to user Y. The time delay in transmitting commands to X and Y would then be 65 milliseconds.

The accuracy of NTP used in broad area network is about 50 milliseconds. NTP computes the time delay based on the back-forth transmission of packets and therefore narrows down the timeframe. Also, the accuracy of Windows timer can reach 1 millisecond. The third method allows the program to obtain the usage of CPU first, then to calculate the delayed time. Hence, it is easier to control the execution time of commands. These are the two main reasons for the decision of using this system for synchronized play.

Fig. 6 demonstrates the time delay using the 3rd method. It must include synchronization inaccuracy and Windows timer inaccuracy in addition to End to End delay. The time delay must be within 100 milliseconds.



Codec Delay	< 1 ms
Packetization Delay	10,20 ms
Queuing Delay	Variable
Serialization Delay	< 1ms
Propagation Delay	< 1ms
De-Jitter Buffer Delay	Variable
Synchronization Inaccuracy	< 50ms
Windows Timer Inaccuracy	~1ms

Fig. 6. Time delay in voice transmission

IV. TESTING RESULTS

The 3rd methods mentioned earlier in the paper were tested to determine which method would generate the shortest time delay. When the delay is more than 100 milliseconds, the voice transmission is affected. The time delay is measured for the lag between receiving the voice packet and the length of song being played. The delay is also measured between the start of singing and the time of hearing the voice from the receiving end.

The test is performed on computer X that plays karaoke and on computer Y that receives the first voice packet to measure the time consumed. Computer X is the Mixing Agent while Computer Y is the Singer Agent. The voice packet size is 233Bytes and voice sampling time is 20 milliseconds.

Computer X is located in National Chung-cheng University with Server in Chang-gung University and NTP Server in National Chung-cheng University. Computer Y uses ADSL, and its upload speed is 64kbps and download speed is 1Mbps.

Both Computer X and Y are on the Internet and Computer X, Server and NTP Server are on TANET. Testing equipment specifications are shown in Table II.

TABLE II

TABLE OF TESTING EQUIPMENT SPECIFICATIONS

Role	IP	Operation System	CPU Speed	RAM Size (MB)
Computer X	140.123.113.XXX	WinXP	Pentium 3G	512
Computer Y	61.70.69.XXX	WinXP	AMD xp2000	512
Server	163.25.101.XXX	WinXP	Pentium 1G	256

Fig. 7 shows the testing environment of the 1st method. Testing steps are Computer X sending commands to Computer Y and then Computer sending voice packet to Computer X after receiving commands. The time measured is from the millisecond when Computer X sends out commands to the millisecond when Computer X receives the first voice packet sent from Computer Y.

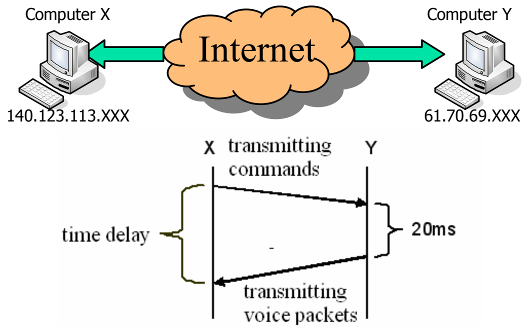


Fig. 7. Testing environment of 1st method

Fig. 8 demonstrates the testing environment of the 2nd method. The procedure is that the Server sends commands to Computer X and Y. And Computer Y sends voice packets to Computer X after receiving commands. The time measured is from Computer X receiving commands from Server to Computer X receiving voice packets from Computer Y.

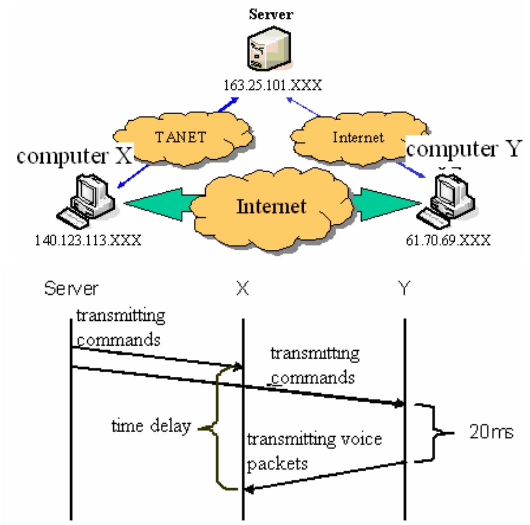


Fig. 8. Testing environment of 2nd method

Fig. 9 shows the testing environment of the 3rd method. After Computer X and CCU NTP Server are time-checked, Computer X sends commands to Computer Y. Then upon receipt of commands, Computer Y is time-checked with CCU NTP Server. Computer X and Y send out voice packets to each other at the same time. The delay time is measured from the wake-up of

Computer X to the receipt of the first voice packet from Computer Y.

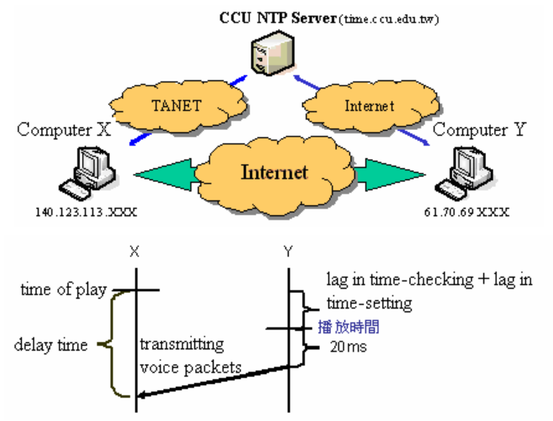


Fig. 9. Testing environment of the 3rd method

By using the 1st method, the testing result of delay time shown in Fig. 10 is more than 100 milliseconds due to the longer transmission time on the internet. By using the 2nd method, the result shown in Fig. 11 is less than 100 milliseconds. The testing result of the 3rd method shown in Fig. 12 is also within 100 milliseconds, but the time delay is smaller than that of the 2nd method. Table III shows comparison of this system with the other popular online karaoke systems.

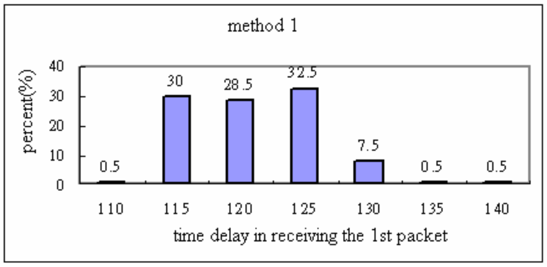


Fig. 10. Time delay in method 1

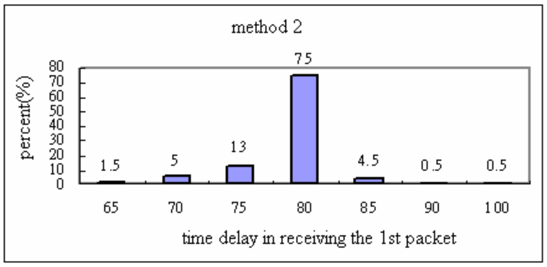


Fig. 11. Time delay in method 2

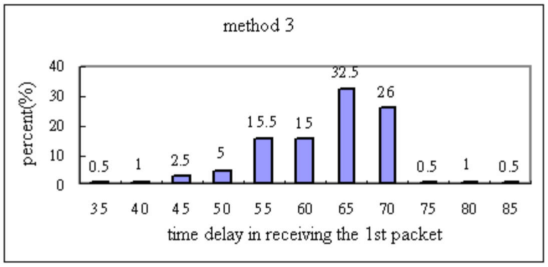


Fig. 12. Time delay in method 3

TABLE III
COMPARISON OF THIS SYSTEM WITH THE OTHER POPULAR ONLINE KARAOKE SYSTEMS

Function	This system	Karaokefan	Yahoo KTV
free	○		○
songs		○	
Solo singing	○	○	○
Multiple singing	○		
Microphone switching	○		
Voice recording	○	○	○
chatting	○	○	○
Automatic song –listing	○		
Order matching	○		
Public service	○		
Private service	○		
service management	○		
Instant scoring	○		
Data searching	○	○	

V. CONCLUSION AND FUTURE DISCUSSION

This paper proposed the design of CFAS with multi-singing karaoke system, in which the skills and mechanisms are detailed devised and simulated. For the instant interaction application, the bearable delay is within 100 milliseconds. Hence, the basic requirement for instant singing application is that all singers must start simultaneously and be within 100 milliseconds. The method used is to let all singers time first. Then we synchronize the play of the karaoke to accomplish multiple singing at the same time.

There is room for further development of this system and there are some issues worth discussing and researching.

1) Copyright: If more songs are available, more users will be drawn to use this system. To solve the copyright problems while increasing song selection is an important direction for the future.

2) Complete scale: Having the web-camera function will allow users to not only hear the voice but also see the images of themselves and other performers. Providing web private space will also allow users to save their own voices and other users feedback which will increase the attraction of friend-making.

ACKNOWLEDGMENT

This paper was supported by the National Science Council under contract number NSC- 95-2219-E-194-005-. In addition, the author is very grateful to the relevant researchers' supports and the anonymous reviewers for their suggestions and comments.

REFERENCES

[1] Helmut Haas, "Über den Einfluss eines Einfachechos auf die Hörsamkeit von Sprache", University of Gottingen, Germany, December 1949.

[2] Xiaoyuan Gu, Matthias Dick, Zefir Kurtisi, Ulf Noyer, and Lars Wolf, Technische Universität Braunschweig,"Network-centric Music Performance: Practice and Experiments", IEEE Communications Magazine,vol. 43, no. 6, pp 86-93, June 2005.

[3] Cheng Hung-yu, VOIP network telecom techniques, Sung-gang publishing company. Aug. 2005.

[4] D. Mills.," Simple Network Time Protocol (SNTP)", IETF RFC1769, March 1995.

[5] How To: Time Managed Code Using QueryPerformanceCounter and QueryPerformanceFrequency, [online].Available: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag/html/scalenethowto09.asp

[6] Abraham Silberschatz, Peter Bear Galvin, Greg Gagne, " Operating system concepts,6th ed.", Wiley, 2002/9.

[7] D. Mills.," Network Time Protocol (Version 3) Specification, Implementation and Analysis", IETF RFC1305, March 1992.

[8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson., "A Transport Protocol for Real-Time Applications", IETF RFC1889, January 1996.