DESIGN AND IMPLEMENTATION OF A DESKTOP COMPUTER SUPPORTED COOPERATIVE WORK SYSTEM[1]

Tsang-Min Chen, Chun-Chuan Yang, Wei-Hsin Tseng, Ing-Chau Chang, Jau-Hsiung Huang,
Chi-Chang Lin, Mong-Shu Lee, Nie-Jiang Fon, and Sheily Kaw
Communications and Multimedia Lab.
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

In this paper, we describe the implementation of a desktop CSCW (Computer Supported Cooperative Work) system using commercially available consumer electronic devices and facilities. The system is developed on workstations under UNIX and OPEN LOOK environments connected by FDDI (Fiber Distributed Data Interface) networks. It provides multi-party conference with shared white board, shared text editor, calendar, and voting capability. Each conference with these features can support a groupware environment. The design and consideration of the framework of the system is presented. During the implementation, some performance issues, such as error control, medium synchronization and throughput improvement will also be discussed.

1. Introduction

Due to the increasing power of computer computation and high speed network technology, more services can be provided than a video conference system to support cooperative work environment. A CSCW system is hence implemented to meet this demand. This paper describes how video conference systems [1-7] can be enhanced to become a CSCW system.

Previously, we have implemented a video conference system with shared white board and some design limitations and considerations were discussed in [1,2,7]. In designing a CSCW system, network design considerations and the management to support traffic of different media with real-time constraints are the key issues. The system presented here provides a desktop environment to achieve CSCW using commercially available consumer electronic devices.

The system concept is shown in Fig.1 where a session manager is used to maintain the overall CSCW system. In addition to session manager, there are also voting agent and calendar agent to support coordination between members in each conference. Both voting and calendar agents can not be directly accessed by all members, including chairmen and attendants, in order to avoid human intervention. The information of each conference includes conference name, type, member list, and date. The conference type will be later explained in section 2.2. These information are provided by the chairman before each conference. Each conference is initiated at the same time as its chairman is registered. After a conference is registered by its chairman, the control of these information are transferred to the chairman. Therefore, each conference can progress independently with others.

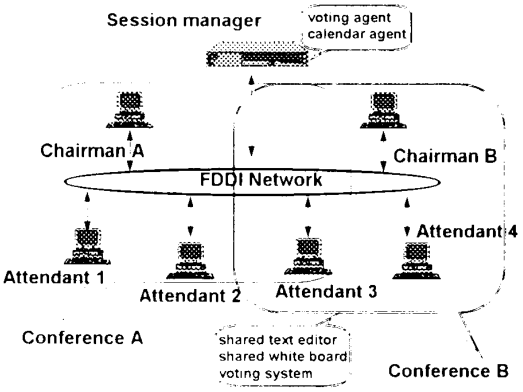


Fig. 1. The CSCW architecture.

[1] This work was supported by The National Science Council of ROC under grants NSC-83-0408-E-002-003, NSC-83-0408-E-002-005, and NSC-83-0425-002-140.

Revised manuscript received September 14, 1994 0098 3063/94 $04.00 © 1994 IEEE

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

Each attendant can dynamically register in multiple conferences which are controlled by their chairmen, respectively. The monitor display of each attendant contains up to five video windows with a resolution 360*240 each. Video data is compressed using JPEG (Joint Project Expert Group) boards; however, a software video codec [7] can also be used to reduce the cost. Each attendant can dynamically switch the context of any window to available remote video sources from different conferences. The frame rate of video ranges from 6 to 12 frames per second depending on the number of video windows opened. Note that the video resolution, number of video windows opened, and frame rates are higher than those in [1,2,7].

Two formats, compressed form and vector form, are used to transmit the data of the shared white-board depending on the performance and network utilization. The compressed form, compressed by a 2-dimensional run-length coding, is used to transmit the entire screen in cases of loading the screen or joining of new attendants. In addition, the drawing and controlling commands are transmitted using vector form which are organized by drawing tools and commands. Transaction based locking mechanism is used to coordinate shared text editor based on each paragraph of text files. Several attendants can update different paragraphs simultaneously after obtaining the permissions.

Other than the above shared environment, a voting mechanism and a calendar are needed for a CSCW system. The voting capability is provided to resolve conflict of opinions. The key design issue is how to obtain justice and anonymity. Each voting activity is created by the chairman and then voting agent will sent it to all attendants who have registered in the conference. Each attendant can obtain the voting results(voting count) automatically from the agent. The calendar is used to coordinate members' schedule. It can automatically detect time conflict between members' schedule in arranging a conference meeting. It will also send mails to each member to announce the arrangement of the conference meeting.

There are many design issues to be taken into account in implementing such a system. Some issues have been discussed in [13,14]. This paper describes how multimedia conference systems can be enhanced to become a CSCW system during the implementation. In section 2, we show the system architecture first and show the design and implementation considerations in section 3. Finally, we conclude some remarks in section 4.

2. System Architecture

The hardware of the system contains a group of workstations. On each workstation, there is a desktop environment, connected by an FDDI network, which provides friendly GUI (Graphical User Interface) to users. The GUI is supported by OpenLook widget programming and X-window Programming on X11R5/Openwin3.0 environment. Beside these, a motion JPEG compression board is required to grab and compress video frames[8-10]. For audio processing, the workstation has built-in functions which support μ _law PCM (Pulse Code Modulation) audio encoding scheme. Moreover, a hidden microphone and a speaker are also built in the workstation as audio devices.

Due to characteristics of each medium, both TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) multicasting are used in the system. Different constraints on error control are provided. These usage will be discussed in section 3. We will first describe the system modules as shown in Fig.2.

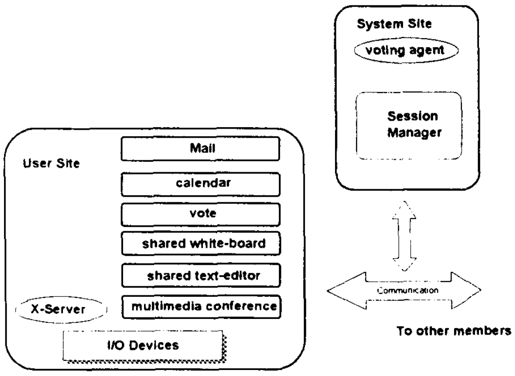


Fig.2. The CSCW modules.

2.1 Session Manager

The session manager, shown in Fig.3., is the soul of the system. It maintains all information for the initiation of each conference. The information include name, type, members, and dates. Each Chairman can connect to the session manager remotely to setup/cancel the conference, add/delete a member, and modify the right of each member. After initialization of a conference, the database is used only for querying the status of each conference. The maintenance and control of the progress of the conferences are then transferred to each chairman.

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

On the window of session manager, each icon represents a conference set up by its chairman. We can modify all features of a conference using friendly GUI. For each conference, all members are shown with their rights on the window. The rights include vote, video, audio and an indication to show whether the member is still on-line. Beside these, the voting agent is also implemented in the session manager which provides the voting capability. This feature will be discussed in Section 2.5.

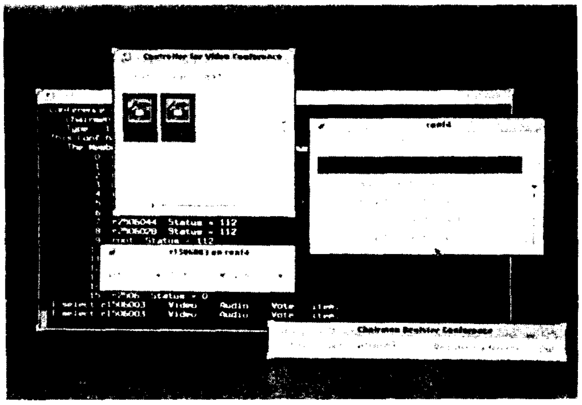

Controller for Video Conference
conf4
r1506003 on conf4
Chairman Register Conference
r2506044 Status = 112
r2506020 Status = 112
root Status = 112
r2506 Status = 0
select r1506003 Video Audio Vote item
select r1506003 Video Audio Vote item

Fig.3. The display of session manager.

2.2 multimedia conference

There are many multimedia conferences[2,5-7] implemented in difference platforms. Our system is implemented on an FDDI network since the bandwidth requirement of a single video stream is about 4.8Mbps (5 video sources * 12 K bytes per frames * 10 frames per second * 8 bits per byte = 4.8 Mbps). High speed trunks (e.g. DS3 link) can be used to connect different FDDI networks for a wider area coverage.

The display of the monitor is shown in Fig.4. It includes a multimedia conference, a shared-white board (on the lower-right part) and a shared text-editor (on the lower-left part). There are five video windows shown in a conference. One of the windows is the video of local site and the others are remote sources. Shared memory implementation is used to improve video grabbing and display. The overall efficiency is limited by the speed of video grabbing. From our experience, the time for video grabber to grab a frame is about 40 ms/frames. Due to the computation power of the hardware, our system can provide a video rate up to 12 frames per second in the case that the number of the attendant is smaller than 5. Although the total number of attendants may be larger than the number of video windows, each attendant can dynamically switch the context of video window to available remote video sources.

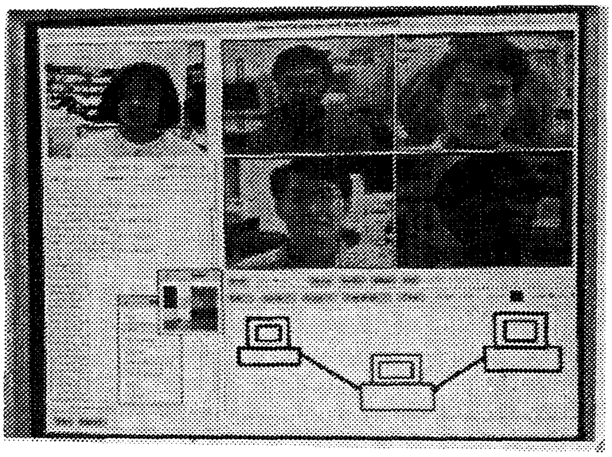


Fig 4. The display on user site.

The system also provides multi-party conferences. An attendant can dynamically register to a conference after obtaining the permission from its chairman. After registering to conferences, an attendant can select one of the registered conferences to join at any time. However, the system does not allow multiple conferences to be displayed on one user site simultaneously. Similarly, a chairman can not register to multiple conferences since it may cause problems in conference controlling.

Two types of conference room are supported in our system. One uses a token to indicate the current speaker in the conference room. This token can be assign by the chairman. The attendant can speak after obtaining the permission of audio from the chairman. Another type of conference room supports audio mixer capability. Due to synchronization of audio and video, voices of all members are mixed together at the receiving site instead of at the sending site.

The continuous playing of audio data and the synchronization between media are also achieved by skillful design using two levels of module scheduling to improve the overall efficiency. This will be discussed in Section 3.

2.3 Shared white-board

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

For shared white board, it provides each attendant to use the keyboard or mouse to draw pictures with a resolution of 720*240. Various colors and functions can be selected from the toolbox to share drawing picture with others. The toolbox includes color, pen width, and shape functions, etc.. Two types of data have to be processed in the shared white board. In normal situation, each attendant can use the keyboard or mouse to draw picture on it. The shared data has to be multicasted to all others using vector form which can be translated to the command and operation at the receiving site.

In another case, the entire content has to be transmitted while new attendant takes part in. Chairman uses point-to-point transmission to transmit the entire screen of the shared data to the new attendant. This data (almost 100kBytes for the entire shared white-board screen) should not be transmitted directly without any compression due to performance issue. By real-time constrains, a 2-dimensional run-length codec provides fast and adequate compression rates. These shared data will then be transmitted in the compressed form and required error-free transmission. Furthermore, compressed form of white-board data is also needed while a picture is loaded and saved.

2.4 Shared text-editor

As well as the multimedia conference and the shared white board, shared text-editor also plays a major role in CSCW system. Shared text editor is controlled by the chairman using transaction operation. Each transaction contains a paragraph of shared text which can be updated by an attendant after being locked. Each text file is shared in a conference only. Chairman can load/save a text file and provide a lock scheme related to each paragraph.

On the lower-left part in Fig.4, the window provides the system message or shared editor. In normal situation, window will display the information of the system status. After loading a text from the chairman, all members can modify the text simultaneously from this window after obtaining the permission. Due to the performance constraint, the updates will be based on the entire paragraph instead of on characters. Therefore, temporary inconsistency may occur during the updates. After the update is confirmed or canceled, chairman will send the shared text to all attendants. The consistency is then achieved.

2.5 Voting system

In CSCW, voting capability is also required to support decision making. To prevent human intervention, the agent should not be accessible by any user. Chairman can connects to the voting agent to create the single-answer or multiple-answer vote. The voting agent will then send the ballot to each attendant and will pop-up a decision window on the monitor of each user. After an elapsed time decided by the chairman, the vote count (result) will be sent to each attendant automatically. Fig.5. shows the display of the voting ballot and voting result on user sites.

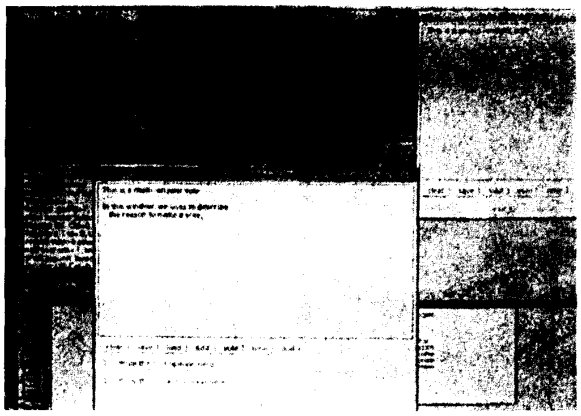


Fig. 5. The display of voting system.

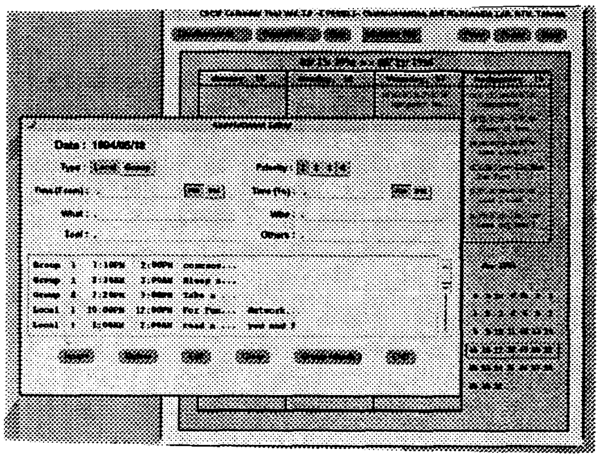


Fig. 6. The display of calendar system.

2.6 Calendar

The calendar agent is used to provide an electronic assistant in CSCW system. It can manage the personal

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

schedule and remind important events to users. It also can automatically start up some programs to be executed if pre-planned. As well as these features, the calendar can communicate with other members' calendars to make an appointment during the progress of a conference. By so doing, it can find some time periods which are available to all members. The time periods are provided with multiple priorities set by each member. If there are no available time periods to all members, the chairman has the right to decide the time for the meeting. In Fig. 6, we show the display of the calendar on user monitor.

3. Design Considerations

There are many design issues to be considered in implementing a CSCW system. We have to support different error control mechanisms to provide efficient usage of resources. To provide different error control considerations, different transmission protocols are used. In this section, we focus on the following issues: 1) how to select an available protocol with respect to a specific medium and how to design error control mechanisms accordingly, 2) how to provide media synchronization and 3) how to schedule video/audio modules to improve system performance.

3.1 Network protocols and error control

In choosing network protocols, a point-to-point protocol, such as TCP, would be impractical to send shared data since each packet has to be sent several times during a conference. Besides, the protocol stack and checksum of TCP are the bottlenecks of the performance when sending real-time traffic, such ad video and audio[14]. Hence, we used a port assigning mechanism together with UDP to achieve the multicasting capability. By so doing, the network traffic load is greatly reduced.

In [11,12], it was pointed out that 100% error free transmission of voice packets and video packets are not necessary as long as they are properly encoded. For example, it has been shown in many experiments that digitized voice packets can suffer a packet loss percentage up to 2%. Therefore both voice and video can be carried without error control under efficient video/audio performance control. Hence UDP multicasting without error control is a reasonable candidate for these media.

However, the connection management information requires error-free transmission since it is the core of the system. For example, if the chairman has to leave the conference early, a new chairman must be elected using the vote capability. Therefore, these features which requires error free transmission can be provided by TCP.

As well as system control information, the data of both shared text and shared white-board need error free transmission to maintain consistency. Due to performance constraint, a multicast protocol has to be implemented to transmit the shared data. Therefore, a UDP multicast with re-transmission protocol is provided here. The UDP multicast protocol is used to transmit the shared data. And, the application program uses a UDP point-to-point protocol to request a re-transmission from its peer application program when errors happen in the system.

To summarize , we use the following table to describe all network protocols and error control mechanisms used in the system :

medium	network protocol	Tx method	real-time	re-Tx by AP
video	UDP	multicast	Yes	No
audio	UDP	multicast	Yes	No
Vector form of SWB	UDP	multicast	No	Yes
Compressed form of SWB	UDP	point-to-point	No	Yes
Text data of STE	UDP	multicast	No	Yes
error re-Tx data sent by AP	UDP	point-to-point	No	Yes
The information of vote, calendar	TCP	point-to-point	No	------
The control and maintenance information of the system	TCP	point-to-point	No	------

Abbreviation :

AP : Application programs

SWB : Shared White-Board

STE : Shared Text-Editor

------- : error free transmission

Table 1 : The network protocols and error control

3.2 Medium Synchronization

Audio and video synchronization can be achieved using interleaved transmission at the sending site and delay/jitter detection at the receiving. The audio packets are used to synchronize other media because there is implicit timing information in it. At the sending site, audio packets are sent periodically with respect to audio sampling rate. At the receiving site, the audio device plays the voice with the speed the same as sender. Therefore the jitter of audio packet which arrives too early can be detected by examining whether the device buffer is empty or not.

To reduce the end-to-end delay of media transmission in the system, which is about 300ms now, we do not want to introduce extra buffers on receiving site. Even if we do not introduce extra receive buffers which can be used to avoid the discontinuity of voice, the delay of voice will still be accumulated by delay jitter of audio packets, especially when the network load is heavy. In the system, we use silent audio packet detection to remove this delay accumulation.

Another method to resolve this delay jitter and accumulation problem can be achieved by the priority scheduling supported by FDDI networks. That is, we can send audio packets with higher priority or send audio packets in synchronous traffic mode, so that the audio packets will suffer much smaller delay and jitter and can be used to synchronize other media.

We use sequence number and synchronization index to detect the mis-ordering of packets and the delay or jitter of media transmission. In the system, we use 8k sampling rate to record audio such that there will be 1k bytes of audio data every 125 ms. That is, there are 8 audio frames (whose size is 1k bytes) and N video frames (whose size is 8-12k bytes after being compressed) to be transmitted per second, where N depends on the video grabbing rate determined by the system, which can also be set by user at its monitor. Audio frames are grouped two by two, where each group is labeled with the same synchronization index. This index is also tagged in the corresponding video frames.

Video frames with the same index are arranged to be sent in the same time interval as their audio counterpart, and are supposed to be received at about the same time as the corresponding audio frames are received. More precisely, the video frames are required to be received in the same time interval as the corresponding audio frames, or either the previous one or the following one time interval. Otherwise, they should be discarded. This provides audio/video synchronization in 500ms. In Fig.7, we show the operation of sequence number and synchronization index. For practical implementation, sequence number and synchronization index can be implemented in the same data field.

At the sending site, each audio frame can be carried in an FDDI MAC (Media Access Control) frame and then be sent periodically. Fig.7a shows that the periodical sending of audio frames with implicit timing information in them. Now, we show the cases which will cause timing and synchronization problems at the receiving site in following figures. Fig.7b shows that the loss of audio packet. In the figure, we use the dash (-) to denote the loss of packets. This loss can be detected by the sequence number and it is treated as a receipt of a silent audio packet. Due to video frame fragmentation, if an FDDI frames is lost, all the FDDI frames belong to the same video frame have to be discarded. In Fig.7c, the last FDDI frame in the video frame 1 is lost. It can be detect by the segmentation index which also can be implemented in the sequence number in the practical system. Then, all FDDI frames in the video frame 1 have to be discarded. The cross (x) in the figure is denoted as discarded frames.

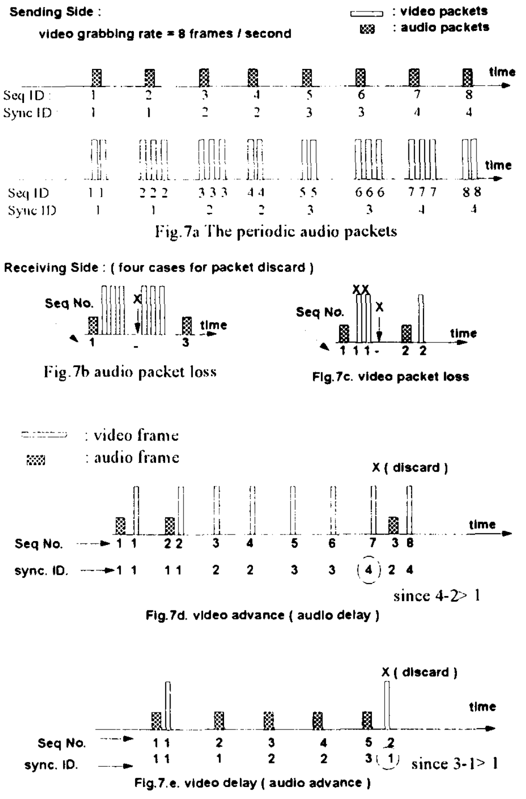


Fig.7a The periodic audio packets

Fig.7b audio packet loss

Fig.7c. video packet loss

Fig.7d. video advance (audio delay)

Fig.7.e. video delay (audio advance)

Fig.7 The synchronization of audio and video.

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

For the audio and video synchronization, we use synchronization index to achieve this constraint and show the examples with the video grabbing rate 8 frames per second in Fig.7d and Fig.7e. In Fig.7d and Fig.7e, we show the cases of audio delay (i.e. video advance) and video delay (i.e. audio advance), respectively. In Fig.7d, video frame 7 has the sync. index 4 which is larger more than one with respect to the next audio packet (its index will be 2). Then, the FDDI frames with sync. index 4 will be discarded due to the difference of synchronization index between video and audio are larger than 1. Note that in these cases, there should be 2 video frames in the same time interval since the video frame rate equals to audio frame rate. In the Fig. 7e, we also show the case that video frame 2 comes later than audio frame 5. Similarly, video frame 2 will be discarded also.

3.3 Routine scheduling

In OPEN LOOK programming environment, the system uses event trigger method to handle the receipt of an I/O event. These events include the receipt of real time media packets, e.g. video or audio, and the receipt of non-real time media packets, e.g. text data or commands required by X-server. For each event, OPEN LOOK will bind a program routine to execute it. In this sub-section, we will show how to schedule the program routines associated to each event in order to improve the overall system performance.

Two levels of scheduling are implemented in our system to provide continuous media transmission, they are intra-routine level and inter-routine level scheduling respectively and will be explained in the following.

Inside the program routine, we can predict the arrival of frames to reduce the waiting time of real time frames. This is called intra-routine level scheduling. For intra-routine level scheduling, we mainly focus our attention on the characteristic that a video frame may be segmented into multiple FDDI frames. For instance, the size of a video frame is about 8K to 12K bytes after being compressed. It has to be segmented into two or three FDDI MAC frames. At the sending side, these FDDI MAC frames are sent at the same time. Hence, we can predict that there are one or two consecutive FDDI MAC frames to arrive after the first FDDI MAC frame. Therefore, after the receiving of a video FDDI frame, we use a loop to receive the following FDDI MAC frames and decide (or predict) whether they are video frames or not.

For inter-routine level scheduling, we focus on the characteristics of real time and non-real time media under OPEN LOOK programming on UNIX multi-tasking environment. In the normal situation, there is only one program routine to be binded on an event to simplify the program design and implementation. In this system, the number of program routines to be binded on an event is related to the traffic load and the number of remote video sources. The traffic load can be decided based on the error rate detected by the system. For example, it is better to bind 3*N+C receive routines to the event of arrival of video frames, where N is the number of attendants and C is the parameter which should be chosen larger for heavier traffic load to achieve better performance. Notice that we multiple N by 3 is due to the fact that there are 3 FDDI frames per video frames.

The benefit of inter-routine level scheduling can be revealed in multi-tasking environment. When an event occurs, X-server needs a little time to cope with it. For example, server will check whether the triggered routine is available and whether the event queue is available. If we only bind a receive routine to a real time medium, FDDI frames will be lost when many video frames come at the same time. This is caused by that receive routine can be binded again to the next real time event only after the completion of its previous one (nonpremptive). Hence, the FDDI frame will be lost when the receive buffer is full.

Therefore, we use dynamic assignment to bind the same receive routines associated to an event incurred by the arrival of each real time medium several times. The number of the routines binded to an I/O event depends on the number of remote video sources and traffic load. Therefore, each receive routine can be binded before the completion of previous event during the I/O waiting of the previous receive routines. And then, the receive routines can be invoked by X-server as soon as the completion of previous one. This dynamic scheduling mechanism can reduce the error rate by more than 20%. Beside, this scheduling is also used to resolve the problem of voice discontinuity since the packet loss rate is lower.

4. Conclusion

With the increasing processing power of inexpensive workstations and high speed network technology, users can actually enjoy themselves in the environment of CSCW. We provide a desktop CSCW system with multimedia conferences, shared white board, shared text editor and voting capability. Three levels of error control, protocol design and module scheduling are also considered with skillful

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

implementation to improve the overall system performance. Finally, such a system can be easily ported to ISDN (Integrated Service Digital Network) networks as long as the network platform is available.

References

[1] J.-H.Huang, W.-H.Tseng, M.-J.Ding, Y-S.Su and L.-C.Wu, "VirtualTalker: An On-line Multimedia Systems on Token Passing Networks," IEEE Trans. on Consumer Electronics, Vol. 39, no. 3, pp. 609-618, Aug. 1993.

[2] J.-H.Huang, C.-C.Yang, W.-H.Tseng, C.-W. Lee, B.-J.Tsaur, L.-K. Chuang, and W.-J. Liu, "Design and Implementation of Multimedia Conference System on Broadcast Networks" Proc. 18th Annual Local Computer Network Conference, Miniapolis, pp.337-341, Sept. 1993.

[3] M. Arango et. al. "The touring Machine System," Communications of the ACM, Vol. 36, No. 1, pp 68-77, Jan. 1993.

[4] P. V. Rangan, and D. C. Swinehart, "Software architecture for integration of video services in the ether phone system," IEEE JSAC vol.9, no. 9, pp. 1395-1404, Dec 1991.

[5] W.H.F.Leung, T.J.Baumgartner, Y.H. Hwang, M.J.Morgan, and S.C.Tu, "A Software architecture for workstations supporting multimedia conferencing in packet switching networks", IEEE JSAC vol.8, no. 3, pp. 380-390, Apr. 1990.

[6] C. Nicolaou, "An architecture for real-time multimedia communication system," IEEE JSAC vol. 8, no. 3, pp. 391-400, Apr. 1990.

[7] H.-C.Huang, J.-H. Huang, J.-L. Wu "Real-Time Software-Based Video Coder for Multimedia Communication Systems", Proceedings of ACM Multimedia 93, pp. 65-74, Aug. 1993.

[8] G.K. Wallace, "The JPEG Still Picture Compression Standard," Communications of the ACM, Vol. 34, No. 4, pp 30-44, Apr. 1991.

[9] MPEG standard draft ISO-IEC/JTC1 SC29 on 22, Nov. 1991.

[10] 1989 CCITT Study Group XV, TD 35, Draft revision of recommendation H.261 : Video codec for audiovisual services at px64 kbits/s, Image Communication, pp. 221-239, Aug. 1990.

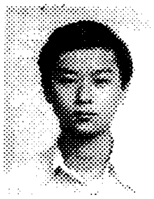
[11] N.S.Jayant and S.W. Christense, "Effects of Packet Losses in Waveform Coded Speech and Improvement Due to an Odd-Even Sample-Interpolation Procedure," IEEE Trans. Communications, Vol. COM-29, pp.101-109, Feb. 1981.

[12] T.M.Chen and D.G.Messerchmitt, "Integrated voice/data switching," IEEE Trans. on Communications, Vol. 26, pp. 16-26, June 1988.

[13] J.-H. Huang and S.-H. Lee, "MHTP : A Multimedia High-Speed Transport Protocol," Proc. IEEE Globecom'92, pp. 1364-1368, Dec. 1992.

[14] J.-H. Huang and C.-W. Chen "On Performance Measurements of TCP/IP and its Device Driver," Proc 17th Annual Local Computer Network Conference, Miniapolis, pp.568-575, Sept. 1992,

Biographies



Tsang-Min Chen received the B.S. and M.S. degree in Electronic Engineering from National Taiwan Institute of Technology, Taipei, Taiwan, in 1990 and 1992, respectively. Since then, he has been a Ph.D. student in the department of Computer Science and Information Engineering at National Taiwan University.

His current research work is focused on high-speed networking, multimedia networking, computer supported cooperative work, and distributed systems.

Chun-Chuan Yang received his B.S. degree in computer and information science from National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., in 1990. He has been a Ph.D. student in the department of Computer Science and Information Engineering at National Taiwan University, Taipei, Taiwan.

His current research topics include multimedia network protocol, admission control, flow control and multimedia medium synchronization.

Wei-Hsin Tseng received his B.S. degree in computer science and information engineering (CSIE) from Tatung Institute of Technology, Taipei, Taiwan, in 1989 and the M.S. degree in CSIE from National Taiwan University (NTU) in 1991. Since then, he has been a Ph.D. student in NTU.

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.

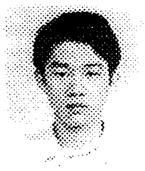
His current research topics include forward correction coding, admission control, video modeling and video server architecture.

Ing-Chau Chang received his B.S. degree in the department of Computer and Information Science from National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., in 1990 and the M.S. degree from National Taiwan University, Taipei, Taiwan, in 1992. Since then, he has been a Ph.D. student in CSIE at NTU.

His current research topics include computer supported cooperative works , network scheduling and distributed multimedia network.

Jau-Hsiung Huang received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981 and the M.S. and Ph.D. degree in computer science from the University of California, Los Angeles, Los Angeles, CA, U.S.A., in 1985 and 1988, respectively.

Since 1988, he has been a member of the faculty in the Department of Computer science and Information Engineering, National Taiwan University, where he is currently a professor. He has published over 40 technical papers in the areas of multimedia networking, high speed networking, parallel and distributed systems and performance evaluation of computing systems.



Chi-Chang Lin received his B.S. degree in Computer Science and Information Engineering from Tatung Institute of Technology, Taipei, Taiwan, in 1993. Since 1993, he has been a graduate student in National Taiwan University. His current research topics is focused on high speed network protocols.

Mong-Shu Lee received his B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1993. Since 1993, he has been a graduate student in NTU. His current research topics is focused on high speed network protocols.

Nie-Jiang Fon received his B.S. degree in computer science and information engineering from TamKang University, Tamswei, Taiwan, in 1993. Since 1993, he has been a graduate student in National Taiwan University. His current research topics is focused on multimedia communication.

Sheily Kaw received her M.S. degree in computer science and information engineering from National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., in 1993. Since 1993, she has been a graduate student in National Taiwan University. Her current research topics is focused on high speed network protocols.

Authorized licensed use limited to: National Taiwan University. Downloaded on March 24, 2009 at 01:49 from IEEE Xplore. Restrictions apply.