ADAPTIVE VIDEO CONFERENCE SYSTEM ON A LOW BIT RATE NETWORK

Fedaa Fahmi Hermez, Prof. Dr. Kais Said Ismail, Ass. Prof. Mr. Hamid M. Ali
University of Baghdad
Electrical Engineering/ Control and Computer Department/ 2007

ABSTUCT

This paper studies video conference system over low bit rate network. As a result of this study, some ideas are employed to transfer video and audio information through the dial up network with minimum jerky views, discontinuous voice and with a reasonable number of users. A video conference standard, which contains video standard and audio standard, is chosen to build a developed Video Conference System that takes into consideration the monitoring of the network load, by using a special designed sensor, called Network Load Sensor. The NLS is built to sense the network load. Based on the test results, the video resolution and audio compression ratio vary continuously. The system is implemented using visual C++, with using of sockets programming as the interface between clients and the conference server and the use of the Windows Multimedia Platform SDK, which support some audiovisual functions.

الخلاصة

يقوم هذا البحث بدراسة نظام المحادثة الصورية عبر شبكات المعلومات الداخلية البطيئة. وكنتيجة لهذة الدراسة ، وظفت بعض الافكار لنقل المعلومات الصورية والسمعية خلال شبكات الهاتف وبأقل معدل من الصور المشوهة والصوت المتقطع وبعدد معقول من المشتركين. لقد تم اختيار معيار المحادثة الصورية، ويضم معيار للصورة ومعيار للصوت، لبناء نظام محادثة صورية مطور ياخذ بنظر الاعتبار مراقبة الحمل الموجود على الشبكة، وذلك بتصميم متحسس خاص، يدعى متحسس حمل الشبكة. تم بناء هذا المتحسس ليتحسس مقدار الحمل الموجود على الشبكة. وكنتيجة لهذا الاختبار، سيتم تعديل ابعاد الصورة ونسبة ضغط المعلومات السمعية باستمرار. ان هذا النظام انجز باستخدام لغة البرمجة Visual C++، و ذلك باستخدام sockets programming للاتصال بين الزبائن و الخادم، وكذلك باستخدام Windows Multimedia Platform SDK لتوفير عدد من الاوامر المتخصصة لمعالجة الصورة و الصوت.

KEY WORDS:
Multimedia Distributed System – Video Conference – Network Load Sensor – H.263 – Frame Rate.

INTRODUCTION

Multimedia involves any combination of two or more of the following elements: text, image, speech, video and applications or programs and Distributed system is a collection of independent computers that appears to its users as a single coherent system [TAN, 02], while, the term Multimedia

Distributed System means the distribution of multimedia processing across a collection of computers in a network.

Multimedia applications like Video Conference VC system generate and process continuous stream of data in real time. They contain large quantities of audio, video, and other time–based data elements, and the timely processing and delivery of the individual data elements is essential. "In distributed system, data transmission is prerequisite, so the main topic in multimedia distributed system is how to transfer multimedia data within the demanded quality"[MIN, 02].

The paper objectives are:

- Study and analyze multimedia distributed systems and take an example the Video Conference system with its standards
- Propose an adaptive system that transmits video/audio in continuous stream with variant resolution, using a dial up system as a media of transmission. The system will not use the internet, in any stage during its operation, as a media of transmission.
- Develop a VC system based on the proposed adaptive system.

The outline of this paper is as follows: section 2, shows the general architecture of any video conference system. Section 3, shows our system architecture and in the section 4 the mechanism of the system is explained. In section 5 the requirements are explained with the performance of the system in section 6, the last section shows the conclusions of the system.

- VIDEO CONFERENCE SYSTEM ARCHITECTURE

The speech and video input are compressed and sent to the communication medium; received audio and video are decompressed and sent to the monitor and speakers. The document information can be displayed together with the video and can be compressed or decompressed in a manner similar to that used for video and audio. When there is not enough bandwidth for communication, audio together with document conference may serve as a viable substitute for VC.

A VC system uses some types of the hardware or software that performs compression is called codec which provides Compression rates of up to 1:500. The resulting digital stream of 1's and 0's is subdivided into labeled packets, which are then transmitted through a digital network of some kind (usually ISDN or IP). The use of audio modems in the transmission line allow for the use of Plain Old Telephone System (POTS), in some low-speed applications, such as video telephony, because they convert the digital pulses to/from analog waves in the audio spectrum range.

A camera and microphone capture the picture and sound of a video session and send those analog signals to video capture adapter board. To cut down on the amount of data that must be processed, the board captures only about half the number of frames per seconds that movies use, which is one reason that the video may look jerky-the frame rate is much slower than the eye is accustomed to seeing. On the video capture adapter card, an analog to digital converter (ADC) chip converts the wavering the analog video and audio signals to a digital format (the only difference for the digital Web Cam is that it has its own ADC and don’t use the video card).

A compression/decompression chip or software (the software is implemented in the digital signal processor (DSP) which takes its instruction from the ROM) reduces the amount of data needed to re-create the video signals.

The compressed video and audio signals are sent to the Network Interface Card (NIC) or modem, which will use the Digital to Analog Converter (DAC) to get the analog format that the telephone line uses through transmission, or may be sent over special telephone lines, such as ISDN or DSL lines, that transmits the data in a digital format without the need to the DAC.

At the remote location a similar PC receives the analog signal by the modem or NIC and converts it through its ADC (the digital signal with ISDN or DSL don't need the ADC), decompresses the incoming signals by the DSP and finally converts the decompressed signals to an analog format to display it on the screen and speakers [HWA, 98] [WIK, 06] [WHI, 99].

* SYSTEM ARCHITECTURE

The proposed system is an application system; it works in the application layer and connects to the network layer through sockets. The system is allocated on a 56 Kbps modem or higher bandwidth; the environment is the telephone line as shown in figure 1, the users arrange the conference time and then establish a connection with the server.

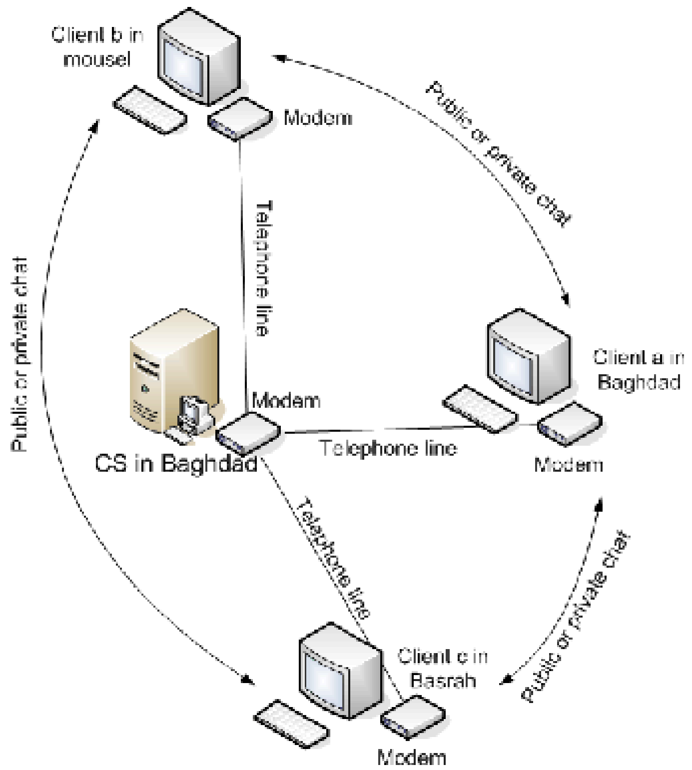

Fig1
The System Connection

Figure 1 shows three clients, each client is in a different city. Those clients may not have an Internet with hi–speed connection to use the existing VC applications. The proposed system offers a VC system with one of the slowest connections existing in these days, the telephone line with modem of 56 Kbps. The three users can have a good conversation as long as the telephone line works. However, each client has to dial the CS which in turn forwards the dial to the intended client (if he is connected) at the other side.

The proposed system works on the telephone line, so it is a local video conference inside one country, for this reason, there is no need for edge servers. Besides, edge servers are not recommended with real time application because they increase latency.

This system is composed of two parts: the conference server (CS) and the clients. The clients are connected to each other through the CS, for this reason and many other reasons mentioned subsection 3.1, the CS is considered to be the administrator of the system.

<u>* The Conference Server</u>

The CS is the part hosting the tables containing information about the clients, PC's Id (Identifier points to the specified client), Rooms' names (room means a number of clients in one group), names of rooms' members and all the clients connected to the system in both (private and public) modules. The main functions of the CS are:

1. **Administrator of the system**: the CS has the responsibility to accept clients and notify other clients about the arrival of new ones. The CS has also the responsibility to organize all the members of a specified room in one list and sends it to all room's members. Also each message sent from one of the members is forwarded to all the members of the same room, in addition to the video of each member.
2. **Error detector of the system**: the CS watches the information packets, if no errors detected, the CS passes the packets after changing some header information to the destination client or clients. In case of detecting an error, like loss of a packet, the server doesn't send an empty packet to the destination; instead the source client observes the absence of the response, so the source client resends the message. This way is more efficient than sending an empty packet to the destination and the latter asks the source to resend it, which is considered a time consuming way.

   When a video packet is lost, the system discards that packet to get a real time VC. Besides, the frame rate of more than 15 fps produces a good video stream that doesn't observe the loss of one packet.
3. **Data storage**: some data are required by the system at any time, so the CS has to store them and updates them continuously. The CS has two types of data storage, one dedicated for the public module which contains the rooms' names, a number for each room and the members of each room. The second storage type is shared between the two types of modules and it contains the nick names of the whole clients and a PC Id number.
4. **Test the network**: the CS is responsible to test the network continuously to sense the load, frame rate and adjusting the video resolution and audio compression accordingly.

Figure 2 shows the main modules that constitute the architecture of the CS:-

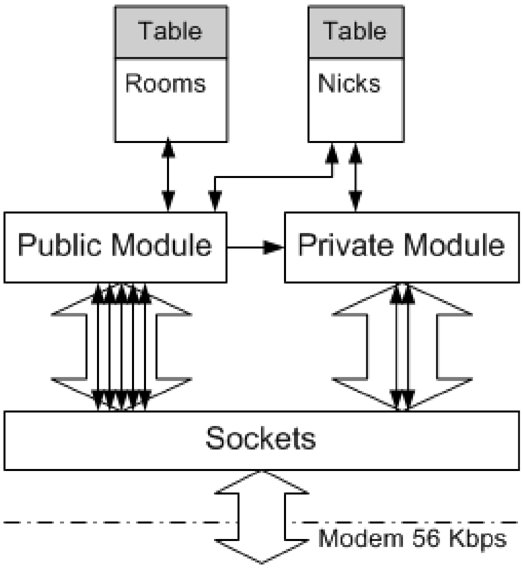

Fig 2
Server Architecture

- ***Private Module***: in the CS, the private module is responsible to pass and watch the normal chat program commands between any two clients through the text socket and audio socket. It is also responsible to change some of the header information.
- ***Public Module***: this part is responsible to pass and watch the audio/video conference commands; through five sockets: one for video stream, one for audio stream, one for NLS, one for text and the last one for the control socket. It also changes some of the header information of the audiovisual packets.
- ***Sockets***: they are objects that represent an endpoint for communication between processes across a network transport. These sockets are created in the private modules to be in the listening mode and then exchange the information with the network after the connection is established.
- ***Rooms' Table***: it's an array storage that contains all rooms created in the system with the members' names of each room and each room has an Id number.
- ***Nicks' Table***: it's another array storage that contains all names of the partner in the system, in both modules, and the PC Id of each computer.

<u>* The Client</u>

The client is another application different from the server in its Graphical User Interface (GUI) and activities; each client opens at least one socket with the CS in order to be connected to other clients. The main functions of the client are:

1. **User Interface**: it represents the interface application that users can deal with to send and receive the multimedia information. It takes the input from the keyboard or camera and displays it on the screen through another window. The audio is taken from the Microphone and playback on the speakers or headphones of another client.
2. **Audio/Video processing**: the client application is responsible to compress/decompress the video stream with many levels of compression, control the audio and process all the required operation of audio/video stream.

Figure 3 shows the main modules that constitute the architecture of the client:-

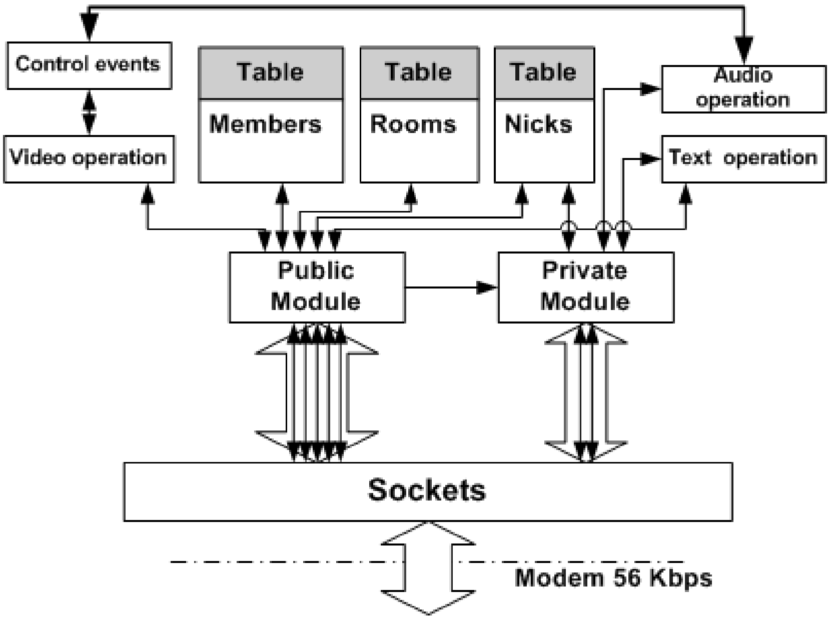

Fig 3
Client Architecture

- ***Private Module***: it's the normal chat program, between two clients, which is responsible to call the text and audio commands. Here the client needs only two sockets and has two lists of the storage (nicks and rooms), but in fact it uses only the nicks list.
- ***Public Module***: it's the audio/video conference program, which is responsible to call all the video and audio commands. When a client is in the public module it may be up to five sockets and it needs all the storage lists.
- ***Nicks' Table***: it's a list of the nick names of each user participated in the system. Also, the nick's table is sent to the client from the CS when the client sends his nick name first.
- ***Rooms' Table***: it's a list of the rooms' names only, sent to the client from the CS when the client sends his nick name first.
- ***Members' Table***: it's an optional list, used only during the public module operation. When a client joins a room, the names of the room's members are sent from the CS.
- ***Text operation***: this operation contains all the functions and variables that are used to send and receive the text information.
- ***Video operation***: this operation contains all the libraries used to capture, send, receive, compress, decompress and display the video information.
- ***Audio operation***: this operation contains all the libraries used to record, send, receive, compress, decompress, and play the audio, the block diagram of the audio and video operations is shown in figure 4.
- ***Control events:*** they are all the event messages, sent by a specified function, used to control audio and video stream. Also, they check audio/video streaming and their connection status.
- ***Sockets***: it is same as in the CS.

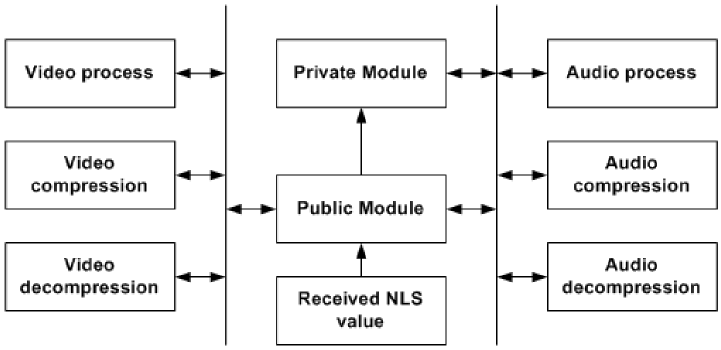

Fig 4
Audio/Video Process Architecture

The following are the main functions of the audio and video operations:

- *Video process*: it's a library that contains all the video operations like: capture, display video on the monitor, send and receive video stream.
- *Video compression*: it's a library that contains the algorithm to compress the video image. The algorithm that is used in the proposed system is the H.263 standard, which is dedicated to the 56 Kbps bandwidth.
- *Video decompression*: it's a library that contains the algorithm of decompressing the incoming compressed video stream.
- *Audio process*: it's a library that contains all the audio operations like: record sound from the microphone, send and receive audio stream and playback the received sound on a speaker or headphone.
- *Audio compression*: it's a library that contains the algorithm to compress the audio stream using the G.723.1 standard.
- *Audio decompress*: it's a library that contains the algorithm to decompress the received audio stream.
- *Received NLS value*: it's a library that contains the value of the NLS received from the CS after testing the network load and select the appropriate video resolution of the clients.

The system uses some of the ready made libraries in the audio/video stream handling like: using of Winmm.lib for the audio and VFW32.lib [MIC, 03] for video, these ready APIs commands are used to capture, send, receive, playback audio and display the video stream.

*** SYSTEM MECHANISM**

The proposed system is a client-server based system. It is a real time human telecommunication using a circuit switching, with a dedicated connection between two terminals for their sole use throughout the session. When a telephone is picked up and dials a number of the server, a circuit is established between the server and the dialed client and it lasts for the duration of the conversation. The same is true for all VC systems and all real-time broadcasting, after the connection is established the information packet can be transmitted.

The main idea of the proposed system is concentrating on the public module, when the user starts to send and receive audio/video information, the proposed system tests the network bandwidth and according to that test the video resolution/audio compression ratio is adjusted. The multimedia information (especially audio/video) requires enormous amount of the bandwidth to transmit its information, for this reason the proposed system uses the H.324 standard that works with telephone lines and applies multi levels of compression for both audio/video data.

Another phenomenon arises when the network load increases. The telephone line is even becomes less than 56 Kbps. The system solves this problem by adapting different resolution levels of H.263 and two levels of compression of the audio using G.723.1.

The proposed system is designed by using a simple sensor, called network load sensor NLS. This sensor tests the network load by creating its own socket for sending a sample packet and count the time it takes to get it back, so it can have the number of frame per second rate. If this rate was less than 15 fps, it can be considered as a high load and if this rate was more than 15 fps that means a low load. This sensor returns its value to the system; the latter increases (if high load) or decreases (if low load) the compression of both audio and video according to the load. Figure 5 illustrates the system flow.

The NLS starts at the CS side, it tests the network every 90 second (fixed period) for a duration of 1 second. The human eye integrates up to about 15 seconds at low light [CLA, 06], but exchanging the video resolution every 15 seconds if the load varies will be unreasonable and the NLS test will going to add additional load on the network. 90 seconds are a good period for eye and don't annoy users. The CS sends about 30 frames to the clients and starts to receive frames from clients for duration of one second. According to the number of the received frames and the previous frame rate, the compression level is adjusted for both the audio and video stream.

There are very rare occasions that the system drops to a point when the load is very high, however the proposed system doesn't hurt the video information because only the video resolution is reduced without affecting the video quality. For audio, increasing the audio compression affects the audio quality, but the received audio considered a reasonable sound with such bandwidth.

The packet that the client sends to server is different in its information depending on why it's sent and by whom, it may contains: the nick name only, message and nick name, message and room name, room name and nick name, video information, audio information or dummy information. The server reads the header information of the packet and changes some of the header information and then forwards the packet to the intended client.

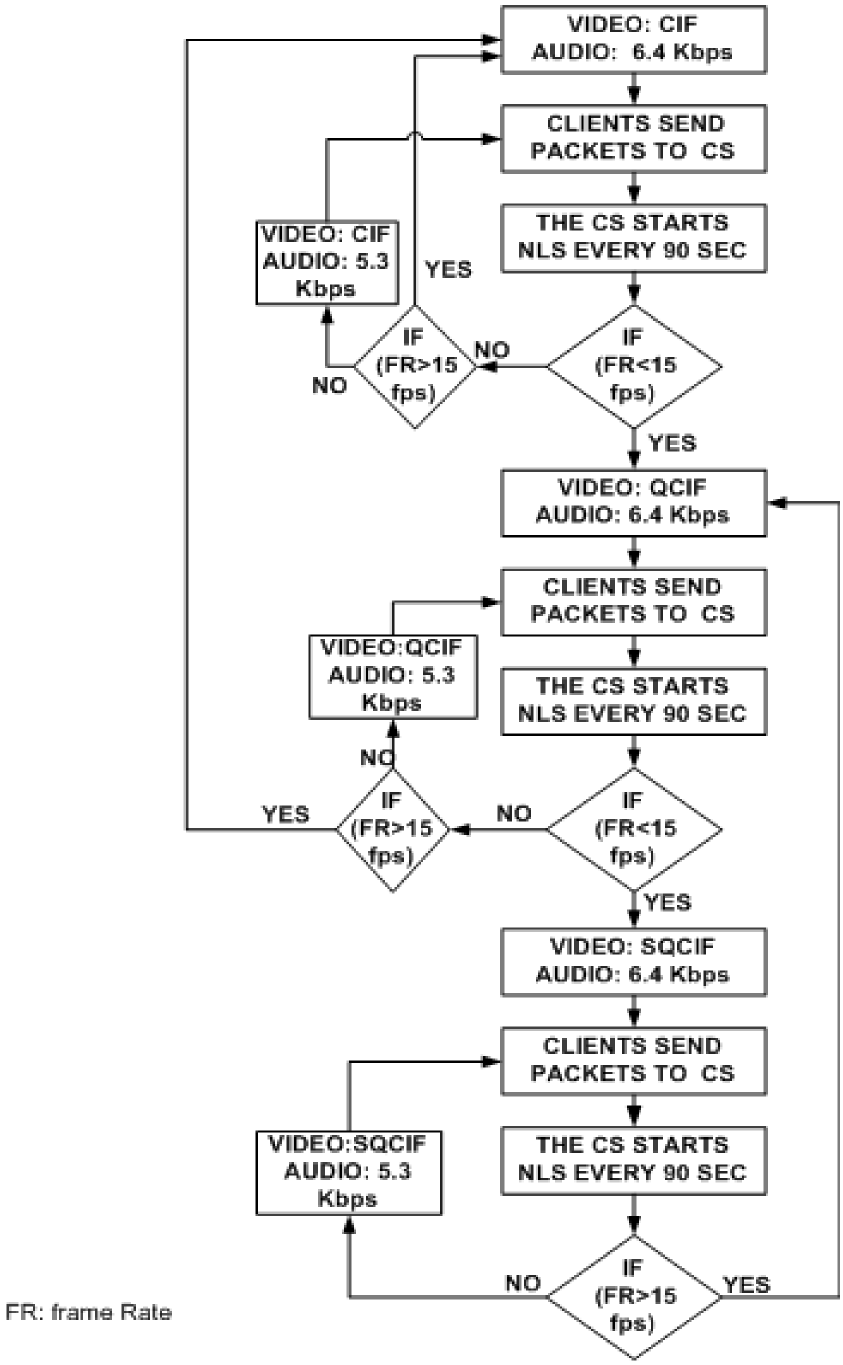

Fig 5
The System Flow

*** SYSTEM REQUIREMENTS**

The client of the system requires windows XP, while the CS requires windows server 2000 or later. The hardware requirements of each client are simple and available almost in any PC. The required equipments for each client are listed below:

- USB WebCam.
- 56 Kbps modem.
- Headphone or speakers.
- Microphone.

The hardware requirement of the CS is only a bank of modems (access server) of at least 100 channels.

- PERFORMANCE RESULTS

The system was tested on PC with the following specifications:

- Pentium IV 1.72 GHz.
- 128Mbytes RAM.
- ATI RADEON Display Adapter.
- 20 Gbytes SAMSUNG Hard Disk.

The image quality is the same for the three resolution levels. Figure 6 shows the QCIF format.

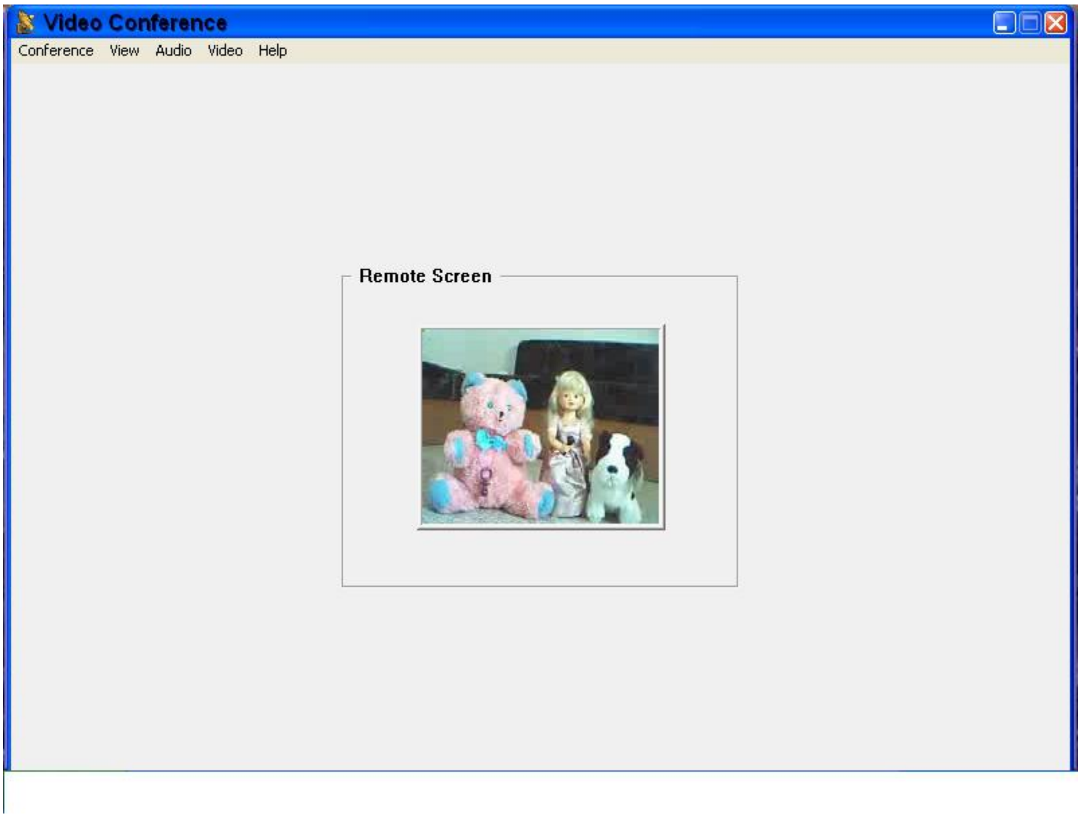

Fig6
QCIF Picture

The performance of the network tested in the laboratory between two clients connected through Bus LAN. Figure 7 illustrates the UDP packets sent and received, Figure 7 A illustrates the network monitor for the CIF format, figure 7 B illustrates the network monitor for the QCIF format and figure 7 C illustrates the network monitor for the SQCIF format.

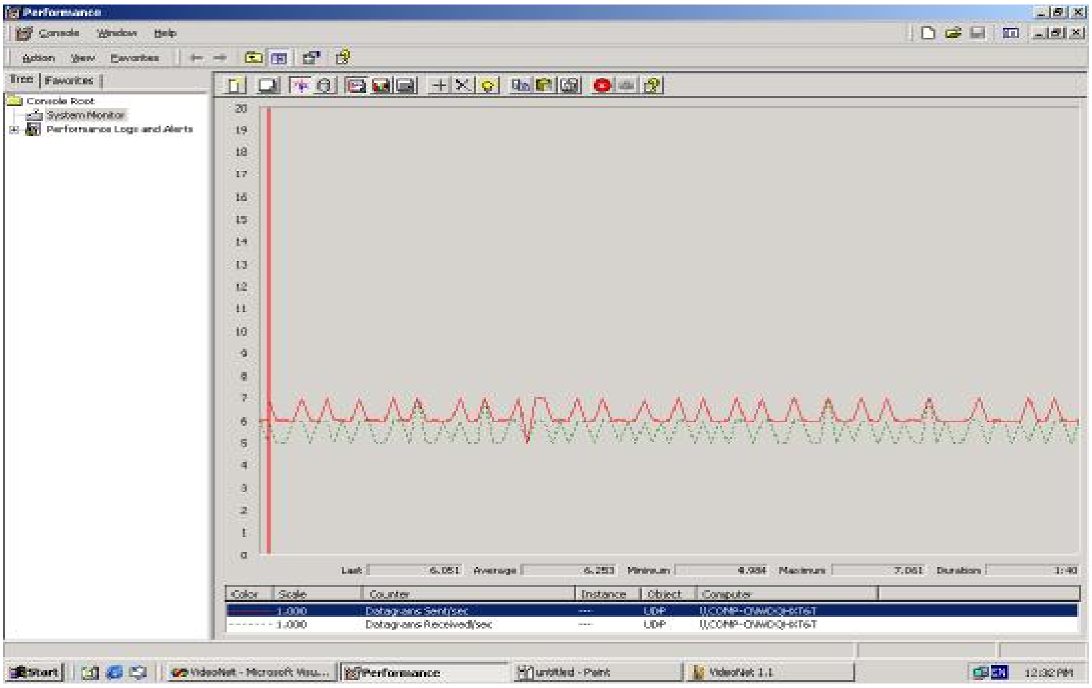

Fig7 A
Network Monitor of CIF

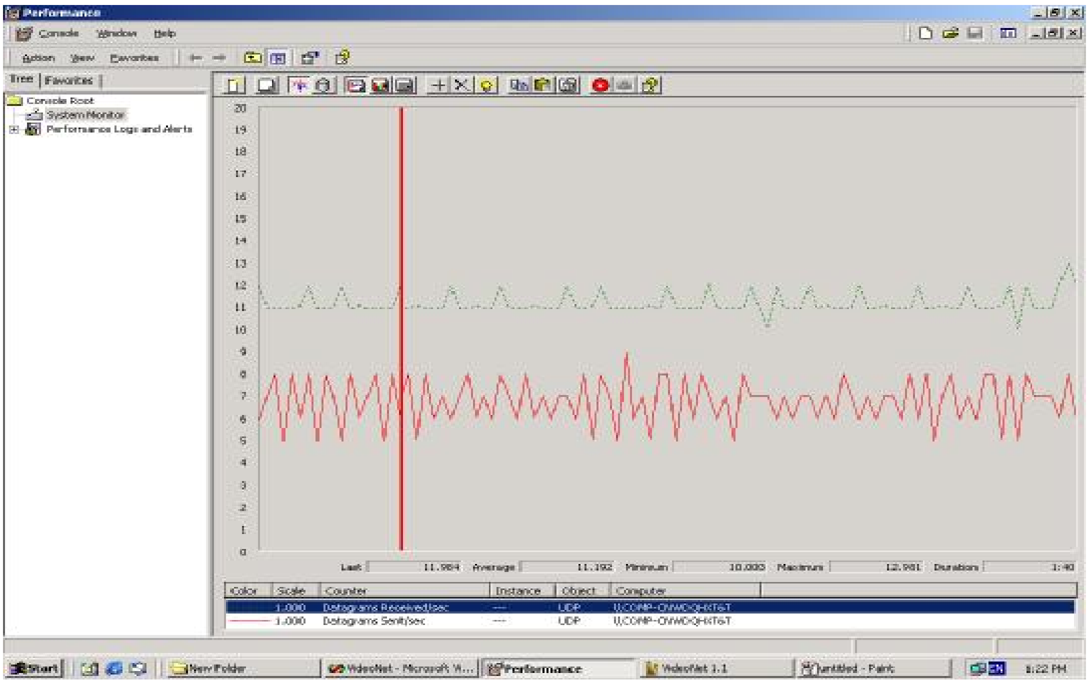

Fig7 B
Network Monitor of QCIF

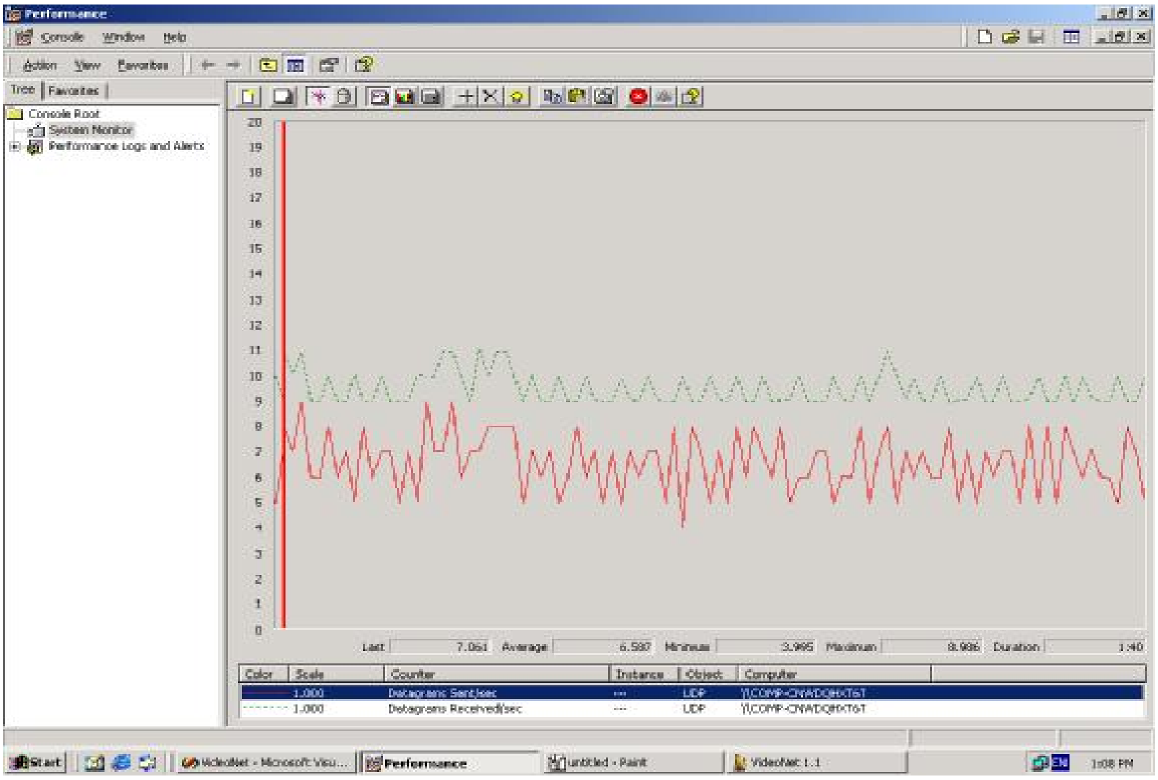

Fig 7 C
Network Monitor of SQCIF

- CONCLUSIONS

During the study and development of the proposed system, several points were observed and noticed.

- When network delay starts to exceed acceptable limits, it is best to reduce resolution of video data, if that reduces delays without making the data unintelligible. The less important information must be degraded first. Video stream should always maintain at minimum 15 fps, frame rate, because this is the limitation that eye perceives jerky motion. Maintaining video at over 15 fps can be achieved by many methods, one of them by adjusting the image size, until it becomes very small.

- Reduce the image resolution, without varying the bandwidth, increases the frame rate.

- Video transmission is more flexible than audio, because it is two dimensional and it is possible to trade off the frame rate with size of image within any required bandwidth.

- Edge servers aren't recommended to be used with video conference systems and any other real time, especially with low bit rate bandwidth, because edge servers are time consuming and must be updated frequently.

REFERENCES

* [CLA, 06] Roger Clark, “Notes on the Resolution and Other Details of the Human Eye”, ClarkVision.com, November 2006, available at: http://www.clarkvision.com/ imagedetail/eye-resolution.html.

* [HWA, 98] Chwan - Hwa and J. David Irwin, “Emerging Multimedia Computer Communication Technologies”, Prentice Hall, Inc., 1998.

* [MIC, 03] Microsoft Corporation, “Video for Windows”, Windows Multimedia, Multimedia SDK, 2003.

* [MIN, 02] Gu Mingyang, “Issues for Distributed Multimedia System ”, Department of Computer and Information Science, IDI, Norwegian University of Science and Technology, 7491 Trondheim, NORWAY, November, 2002, available at: “http://www.idi.ntnu.no/emner/dif8914/essays/Mingyang-essay2002.pdf”.

* [TAN, 02] Andrew Tanenbaum and Maarten Van Steen, “Distributed Systems Principles and Paradigms”, Prentice Hall, 2002.

* [WHI, 99] Ron White, “How Computers Work”, Millennium Edition, Que Corporation, 1999.

* [WIK, 06] Free Wikipedia Encyclopedia, available at: “http://en.wikipedia.org/ wiki/”.

LIST OF ABBREVIATIONS

API: Application Programming Interface
CIF: Common Intermediate Format
DSL: Digital Subscriber Lines
fps: Frames per Second
IP: Internet Protocol
ISDN: Integrated Services Digital Networks
Kbps: kilo bit per second
LAN: Local Area Network
NLS: Network Load Sensor
QCIF: Quarter CIF
ROM: Read Only Memory
SDK: Software Development Kit
SQCIF: Sub QCIF
UDP: User Datagram Protocol
VC: Video Conference
VFW: Video for Windows
Winmm: Windows Multimedia