ISSUES OF MULTIMEDIA DELIVERY FOR E-LEARNING APPLICATIONS

A DISSERTATION

Submitted in partial fulfillment of the

Requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

C SEKHAR KALIVARAPU

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE – 247 667 (INDIA)

JUNE, 2008

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report, entitled **"ISSUES OF MULTIMEDIA DELIVERY FOR E-LEARNING APPLICATIONS"**, being submitted in partial fulfillment of the requirements for the award of the degree of MASTER OF TECHNOLOGY with specialization in INFORMATION TECHNOLOGY, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work carried out from June 2007 to June 2008, under guidance and supervision of **Dr. Ankush Mittal**, associate professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date: 28-6-2008
Place: Roorkee

(C SEKHAR KALIVARAPU)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

Date: 28-6-08
Place: Roorkee

Dr. Ankush Mittal
Associate Professor,
E&CE Department
IIT Roorkee.
Roorkee – 247667, (INDIA)

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report, entitled **"ISSUES OF MULTIMEDIA DELIVERY FOR E-LEARNING APPLICATIONS"**, being submitted in partial fulfillment of the requirements for the award of the degree of MASTER OF TECHNOLOGY with specialization in INFORMATION TECHNOLOGY, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work carried out from June 2007 to June 2008, under guidance and supervision of **Dr. Ankush Mittal**, associate professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

Date: 28-6-2008

Place: Roorkee

(C SEKHAR KALIVARAPU)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

Date: 28-6-08

Place: Roorkee

Dr. Ankush Mittal
Associate Professor,
E&CE Department
IIT Roorkee.
Roorkee – 247667, (INDIA)

ACKNOWLEDGEMENTS

At the outset, I express my heartfelt gratitude to Dr. Ankush Mittal, Associate Professor, Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, for his valuable guidance, support, encouragement and immense help. I consider myself extremely fortunate for getting the opportunity to learn and work under his able supervision. I have deep sense of admiration for his innate goodness and inexhaustible enthusiasm. It helped me to work in right direction to attain desired objectives. Working under his guidance will always remain a cherished experience in my memory and I will adore it throughout my life.

My sincere thanks are also due to rest of the faculty in the Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee, for the technical knowhow and analytical abilities they have imbibed in us which have helped me in dealing with the problems I encountered during the project.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample morale support and valuable suggestions. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

I dedicate this work to my family for their support and encouragement throughout my life.

(C Sekhar Kalivarapu)
M. Tech. (IT)

Abstract

The existing multimedia software in e-learning does not provide par excellence multimedia data service to the common user; hence e-learning services are still short of intelligence and sophisticated end user tools for visualization and retrieval. Delivery of multimedia in the presence of bandwidth constraints is one of the most important video processing problems. The work presents an innovative and complete end-to-end solution for e-learning multimedia delivery package for the client-server model. An efficient approach has been proposed to achieve content based embedded scalable motion compensated video compression, for the e-learning videos, to save and/or efficiently utilize the network bandwidth. A bandwidth effective remote lab monitoring system been designed to reduce bandwidth requirements of virtual laboratory system. An efficient approach to achieve the tasks such as, regional language captioning system and keywords based seek, is introduced.

The work presented in this dissertation will deal with the issue of intelligent transmission of important segments of the video sequence over scarce resource networks. We incorporate the knowledge of the network conditions to determine how various parts of the video frames are encoded. An estimate of the available network bandwidth is obtained which is then distributed optimally between the different frame constituents based on their relative importance and motion by the bandwidth allocation module.

The functioning of Multilanguage content retrieval and management system avails the authoring and displaying of Multilanguage subtitled multimedia content in an effective manner. This system also encompasses a keyword based search and seeks in the authored document with the help of a player. For availing virtual laboratory environment, designed a system that provides remote lab monitoring, doubt resolution, synchronous teacher and student participation. The system acts as an alternative to a laboratory session. Instructor can continuously monitor students and Students can interact with the Instructor at any time they want during the laboratory hours.

CONTENTS

List of Figures and Tables

CHAPTER 1

INTRODUCTION

1.1 An overview

Electronic learning or **E-learning** is a general term used to refer computer-enhanced learning. In many respects, it is commonly associated with the field of advanced learning technology, which deals with both the technologies and associated methodologies in learning using networked and/or multimedia technologies [1]. In the beginning, the primary aim of E-Learning systems was providing flexibility and convenience to the learner. The systems were just an add-on for the university students. But with constant developments in related technologies, another use of E-Learning systems has come up. The new use is to provide quality education to a wide audience, independent of the distance and communication gaps. In comparison with traditional face-to-face classroom learning that centers on instructors who have control over class content and learning process, e-learning offers a learner-centered, self-paced learning environment [2, 3].

Learning has become the general term encompassing the application of computer technologies to education, whether it occurs in face-to-face classrooms, in blended and hybrid courses, in mediated distance education contexts or in online learning environments. Transmission of multimedia data over limited resource networks is in itself a major research issue which has attracted a lot of attention in recent years. Researchers have come up with many solutions to this problem. Most of these solutions rely on increasing data compression ratio and/or optimizing network resources. Though solutions increased efficiency of transmission multimedia data in general, none could be used as such for the transmission of images/videos in e-learning due to the domain specific problems encountered in case of e-learning.

QoS (Quality of Service) for multimedia data transmission [4] depends on many parameters. Some of the most important are, Data rate, Latency (maximum frame / packet delay), Packet loss or error, Jitter and Sync skew. QoS provisioning in wired networks has been the centerpiece of many research activities in packet-switched networks with traffic integration. Since in such networks all types of traffic are transported using common framing which share network resources the transport path, congestion can occur. General QoS has two main purposes, avoid congestion, in case of congestion, and react to it in a way that results in elimination of the congestion over some reasonable time. In wireless and mobile networks, the QoS provisioning problem is even more challenging than in fixed networks.

1.2 Need for Video compression

Digital images and image sequences (video) are a significant component of multimedia information systems, and are by far the most demanding in terms of storage and transmission requirements. The state-of-the-art distance learning focuses on the capture of live lecture presentations in form of lecture videos for subsequent distribution to students at distant locations. By capturing the original presentation through lecture videos one could replicate the same learning experience among distance learners.

Transmission of lecture videos requires a higher bandwidth compared to downloading traditional textual material. The smallest unit of quantization in an image is called a pixel element or pixel. A standard video monitor displays a frame usually with the resolution of 800 * 600 pixels. In color image a pixel is represented by 3 bytes of data. (One for Red, Blue and Green respectively).Thus even one uncompressed image requires 1.373 MB of storage. One hour Video at 15 frames per second will require 72.07 GB of space in our harddisk and is impossible to transmit this leads to need for compression of videos.

One hour of video coded on MPEG standards still takes 500-600 MB of storage. But this is also unsatisfactory. Educational lectures are slow moving videos and specific applications built to compress them on the basis of their content can achieve very high compression standards. In this project we have achieved a high compression rate for educational providing a scalable solution to ensure best Multimedia QoS over various network conditions.

1.3 Need for Bandwidth Adaptive Allocation

The distribution of network resources is generally done statically in traditional multimedia frameworks. However, the network bandwidth keeps on changing. A decrease in network bandwidth will lead to congestion in network while increased network bandwidth implies underutilization of available resources. Dynamic allocation of video content to network according to present network bandwidth ensures optimal utilization of network.

1.4 Motivation

In developing countries, a majority of population dwells in non-urban setup where the educational infrastructure and resources are usually meager and scanty. Trained teaching faculty at primary, secondary and technical educational levels are lacking. The students who come out from such background are less likely to excel than those who are exposed to the best education. However, internet based distance learning (E-Learning) is changing the global scenario occurring in education today. E-Learning techniques [5] – [8] like virtual classrooms provide the exposure to quality education and are very beneficial to students in remote educational institutions. Many institutes such as MIT (USA) and IIT Delhi (India) have opened their web servers for free lecture-on-demand on several courses. Communication and advanced computer technology enable common user to receive instruction despite geographical and time disparity that would otherwise traditional classroom instruction impossible. Nevertheless, the bandwidth restriction on communication channel makes E-Learning a challenging problem. Another major issue of the current E-learning systems is that presently the target audience is very limited. The systems generally deliver educational content in one language only. IN a country like India, with 22 national languages and 1652 dialects, such a system is bound to miss out plenty of audience [9]

1.5 Problem Statement

The goal of this dissertation is to propose multimedia transfer methodologies for E-learning applications.

The primary components are:

1. A system that allows remote lab monitoring, doubt resolution and grading with synchronous teacher and student participation. This requires effective bandwidth utilization between clients and server.
2. A framework that analyses content in e-learning videos and applies specific compression and streaming strategies to deliver the video from server to the client with in the available bandwidth.
3. A Multilanguage content management system for E-learning content. This includes a content definition language and content player.

1.6 Organization of the dissertation

The dissertation is divided into eight chapters including this introductory chapter. The remaining dissertation is organized as follows.

Chapter 2 provides the preliminary knowledge of the various domains used in the dissertation including image and video compression algorithms, bandwidth estimation techniques and image and video streaming strategies.

Chapter 3 gives an overview of the related work that has been presented in the form of literature or working examples. It also discusses the advantages and disadvantages of the previous systems and algorithms and various research gaps

Chapter 4 gives a bandwidth effective implementation of lecture delivery framework, its design and implementation details and issues.

Chapter 5 provides the proposed adaptive lecture delivery framework, its design, implementation and issues.

Chapter 6 presents the Multilanguage content management system for e-learning content

Chapter 7 compiles and presents the various results obtained during the dissertation work, and discusses the pros and cons of the work done based on the findings.

Chapter 8 concludes the dissertation and provides some suggestions for future work.

CHAPTER 2

PRELIMINARIES

2.1 Image Compression

A digital image is a rectangular array of dots, or picture elements, arranged in m rows and n columns. The expression $m \times n$ is called the *resolution* of the image, and the dots are called *pixels* (except in the cases of fax images and video compression, where they are referred to as *pels*). The term "resolution" is sometimes also used to indicate the number of pixels per unit length of the image. Thus, dpi stands for dots per inch. Image compression is the application of Data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Uncompressed multimedia (graphics, audio and video) data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. In general, information can be compressed if it is redundant. It has been mentioned several times that data compression amounts to reducing or removing redundancy in the data. With lossy compression, however, we have a new concept, namely compressing by removing *irrelevancy*. An image can be lossy-compressed by removing irrelevant information even if the original image does not have any redundancy.

2.1.1 Principles behind compression

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS) [10]. In general, three types of redundancy can be identified:

I. **Spatial Redundancy**: Correlation between neighboring pixel values.

II. **Spectral Redundancy**: Correlation between different color planes or spectral bands.

III. **Temporal Redundancy**: Correlation between adjacent frames in a sequence of images (in video applications).

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible.

2.1.2 JPEG:

JPEG (Joint Photographic Expert Group) is a sophisticated lossy/lossless compression method for color or grayscale still images (not movies). It does not handle bi-level (black and white) images very well. It also works best on continuous-tone images, where adjacent pixels have similar colors. One advantage of JPEG is the use of many parameters, allowing the user to adjust the amount of the data lost (and thus also the compression ratio) over a very wide range. Often, the eye cannot see any image degradation even at compression ratios of 10:1 or 20:1. There are two main modes: lossy (also called baseline) and lossless (which typically produces compression ratios of around 0.5). Most implementations support just the lossy mode. This mode includes progressive and hierarchical coding. [11]

This was a joint effort by the CCITT (International Telegraph and Telephone Consultative Committee) and the ISO (the International Standards Organization) that started in June 1987 and produced the first JPEG draft proposal in 1991. The JPEG standard has proved successful and has become widely used for image compression, especially in web pages.

The main JPEG compression steps are outlined below, and each step is then described in detail later.

Step 1: Color images are transformed from RGB into a luminance/chrominance color space. The eye is sensitive to small changes in luminance but not in chrominance, so the chrominance part can later lose much data, and thus be highly compressed, without visually impairing the overall image quality much. This step is optional but important since the remainder of the algorithm works on each color component separately. Without transforming the color space, none of the three color components will tolerate much loss, leading to worse compression.

Step 2: Color images are downsampled by creating low-resolution pixels from the original ones. The downsampling is not done for the luminance component. Grayscale images don't go through this step.

Step 3: The pixels of each color component are organized in groups of 8×8 pixels called *data units* and each data unit is compressed separately. If the number of image rows or columns is not a multiple of 8, the bottom row and the rightmost column are duplicated as many times as necessary. In the non-interleaved mode, the encoder handles all the data units of the first image component, then the data units of the second component, and finally those of the third component. In the interleaved mode the encoder processes the three top-left (#1) data units of the three image components, then the three data units #2, and so on. The fact that each data unit is compressed separately is one of the downsides of JPEG. If the user asks for maximum compression, the decompressed image may exhibit blocking artifacts due to differences between blocks

Step 4: The discrete cosine transform (DCT) is then applied to each data unit to create an 8×8 map of frequency components. They represent the average pixel value and successive higher-frequency changes within the group. This prepares the image data for the crucial step of losing information. Since DCT involves the transcendental function cosine, it must involve some loss of information due to the limited precision of computer arithmetic. This means that even without the main lossy step (step 5 below), there will be some loss of image quality, but it is normally small.

Step 5: Each of the 64 frequency components in a data unit is divided by a separate number called its quantization coefficient (QC), and then rounded to an integer. This is where information is irretrievably lost. Large QCs cause more loss, so the high frequency components typically have larger QCs. Each of the 64 QCs is a JPEG parameter and can, in principle, be specified by the user. In practice, most JPEG implementations use the QC tables recommended by the JPEG standard for the luminance and chrominance image components.

Step 6: The 64 quantized frequency coefficients (which are now integers) of each data unit are encoded using a combination of RLE and Huffman coding. An arithmetic coding variant known as the QM coder can optionally be used instead of Huffman coding.

Step 7: The last step adds headers and all the JPEG parameters used, and output the result.

2.2 Video Compression

Video compression refers to reducing the quantity of data used to represent video content without excessively reducing the quality of the picture. It also reduces the number of bits required to store and/or transmit digital media. Compressed video can be transmitted more economically over a smaller carrier.

2.2.1 MJPEG: Motion JPEG

M-JPEG stands for Motion **JPEG**. M-JPEG is a video format that uses JPEG picture compression in each frame of the video. Frames of the video don't interact with each other in any way (like they do in MPEG-1, MPEG-2, etc..) which results in much bigger file size, but in other hand, it makes the video editing easier because each of the frames has all of the information they need stored in them.

Motion JPEG uses intraframe coding technology that is very similar in technology to the I-frame part of video coding standards such as MPEG-1 and MPEG-2, but does not use interframe prediction. The lack of use of interframe prediction results in a loss of compression capability, but eases video editing, since simple edits can be performed at any frame when all frames are I-frames. Video coding formats such as MPEG-2 can also be used in such an I-frame only fashion to provide similar compression capability and similar ease of editing features.

Using only intraframe coding technology also makes the degree of compression capability independent of the amount of motion in the scene, since temporal prediction is not being used. However, although the bit rate of Motion JPEG is substantially better than completely uncompressed video, it is substantially worse than that of video codecs which use inter-frame motion compensation such as MPEG-1.

2.2.2 MPEG: Moving Picture Experts Group

MPEG (pronounced M-peg), which stands for Moving Picture Experts Group, is the name of a family of standards used for coding audio-visual information (e.g., movies, video, music) in a digital compressed format. The major advantage of MPEG compared to other video and audio coding formats is that MPEG files are much smaller for the same quality. This is because MPEG uses very sophisticated compression techniques.

Frame Packaging: Video compression system identifies three types of frames, depending on the location of all reference block, I-frame (Intra-coded frame), P-frame (predictive frame), and the B-frame (bi-directionally predictive coded frame). The video frame is efficiently coded for the I-frame by motion estimating within the same frame (to reduce spatial redundancy); whereas, the location of target block for motion estimation/compensation decides the type of coded frame (P-frame or B-frame). P-frames are coded relative to a temporarily preceding I or P frame and B-frames are coded relative to the nearest previous and/or future I and P frame. Reasons for coding B-frames are: 1) better matching of blocks, as it can be matched in reverse as well as forward direction, hence lower bpp (bits per pixel) after compression and 2) B-frames can be filtered out from the video stream for lowering the frame rate to improve the load condition of under-performing network, without considerably affecting the video quality at client side. Figure 2.1 shows the relation between I, P and B frames.

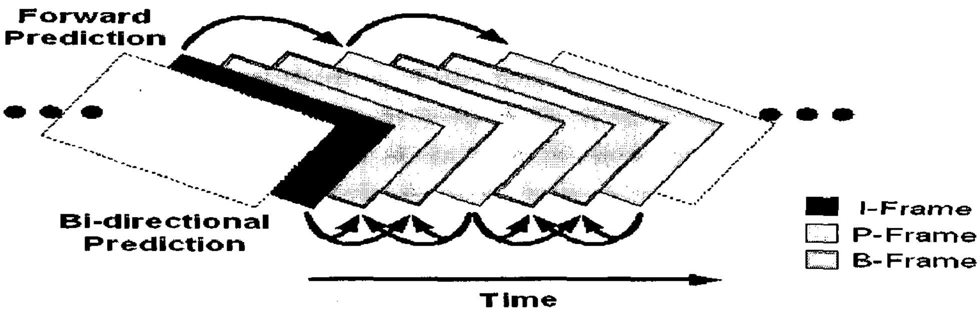

Figure 2.1: Temporal relations between I, P and B

Intraframe vs. interframe compression: One of the most powerful techniques for compressing video is interframe compression. This works by comparing each frame in the video with the previous one. If the frame contains areas where nothing has moved, the system simply issues a short command that copies that part of the previous frame, bit-for-bit, into the next one. If objects move in a simple manner, the compressor emits a (slightly longer) command that tells the decompressor to shift, rotate, lighten, or darken the copy -- a longer command, but still much shorter than intraframe compression. Interframe compression is best for finished programs that will simply be played back by the viewer. Interframe compression can cause problems if it is used for editing.

Motion Estimation and Motion Compensation: In image and video coding schemes image is divided into small blocks for operation by prediction techniques. Motion estimation is used to determine the movement of a macroblock from the reference frame to the current frame. Motion is estimated by searching for the macroblock in the reference picture that provides the closest match. The difference between the values of both the macroblocks is coded for reconstruction at the decoder. To reduce the distortion between the decoded and the original picture, the encoder uses a reconstructed reference frame to perform motion estimation. This reconstructed reference frame is same as used at the decoder side. Motion estimation computes one motion vector per macroblock. Usually, the search is conducted for the luminance component only. A predictive frame is constructed from the motion vectors obtained for all macroblocks in the frame, by replicating the macroblocks from the reference frame at the new locations indicated by the motion vectors. The difference between the values of the predicted and the current frames, known as predictive error frame (PEF), is then encoded using the same procedure as for an intracoded frame. The frame obtained by adding the predictive frame to the PEF is known as the reconstructed frame. The energy of the PEF is low, thus many coefficients are zero, reducing the number of bits needed to encode the frame. Figure 2.2 shows motion estimation of a block.

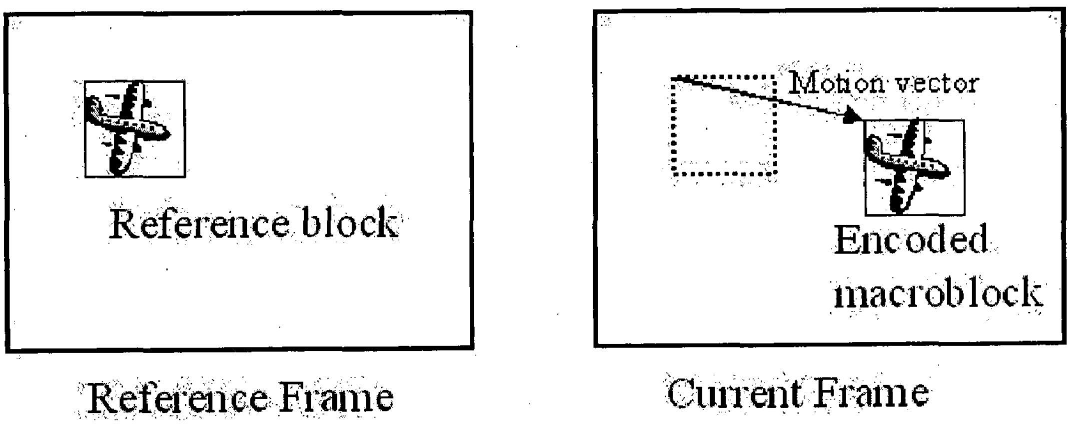

Figure 2.2: Motion estimation of a block

2.3 Bandwidth estimation

In physical layer communications, the term bandwidth relates to the spectral width of electromagnetic signals or to the propagation characteristics of communication systems. In the context of data networks, the term bandwidth quantifies the data rate that a network link or a network path can transfer. The concept of bandwidth is central to digital communications, and specifically to packet networks, as it relates to the amount of data that a link or network path can deliver per unit of time. For data-intensive applications, such as the video streaming system discussed in this dissertation, the bandwidth available to the application directly impacts application performance.

The available bandwidth of a link relates to the unused, or "spare", capacity of the link during a certain time period. So even though the capacity of a link depends on the underlying transmission technology and propagation medium, the available bandwidth of a link additionally depends on the traffic load at that link, and is typically a time-varying metric. At any specific instant in time, a link is either transmitting a packet at the full link capacity or it is idle, so the instantaneous utilization of a link can only be either 0 or 1 [12].

2.4 Multimedia Networking Issues

Multimedia networking issues play an important role in our system. For the purpose of bandwidth adaptive compression we need to estimate the available bandwidth. Also after video compression is complete we need to transmit the compressed video over a network. The choice of networking protocol significantly affects our system performance. Next we give an introduction of Real Time Protocol (RTP) [13] which we have used in our project.

The **Real-time Transport Protocol** (or **RTP**) defines a standardized packet format for delivering audio and video over the Internet. It was developed by the Audio-Video Transport Working Group of the IETF and first published in 1996. RTP does not have a standard TCP or UDP port on which it communicates. The only standard that it obeys is that UDP communications are done via an even port and the next higher odd port is used for RTP Control Protocol (RTCP) communications. Although there are no standards assigned, RTP is generally configured to use ports 16384-32767. RTP can carry any data with real-time characteristics, such as interactive audio and video.

The services provided by RTP include:

a. Payload-type identification - Indication of what kind of content is being carried
b. Sequence numbering - PDU sequence number
c. Time stamping - allow synchronization and jitter calculations
d. Delivery monitoring

The protocols themselves do not provide mechanisms to ensure timely delivery. They also do not give any Quality of Service (QOS) guarantees. These things have to be provided by some other mechanism.

2.5 Video streaming

Streaming video has become a practical technology due to advances in streaming technology itself, coupled with the increased penetration of broadband Internet access throughout the country. It is a technology that is generally useful in education. It provides tools for enabling and enhancing course delivery not only for online students, but also for traditional students attending on-campus classes.

Delivery of streaming video to the client makes use of one of two general models. The stream is either delivered live in real-time, or is made available on client demand from an archived file. There are a few differences in quality and delivery mechanics between the two models. Some minor quality issues are introduced indirectly in live delivery due to the constraints imposed under a real-time production model. These constraints are absent in the creation of an archive file. For example, the production of streaming files for on-demand viewing allows for finer-grain and multi-pass optimizations. Other differences lie in the specific mechanics used for delivery to the client. Live, real-time streams are delivered using a media server. On-demand archives can be delivered either using a media server, or by using an Internet server.

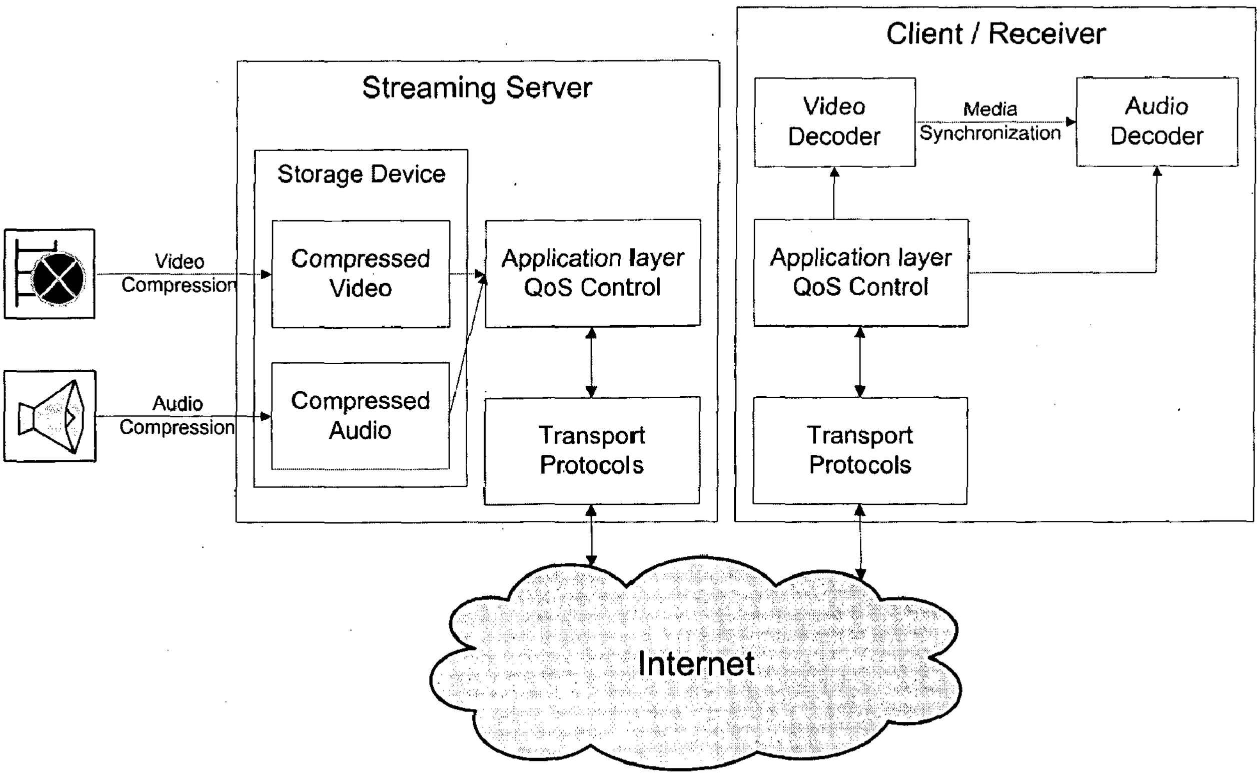

Figure 2.3: Basic architecture for video streaming.

Basic architecture: The basic architecture of any Video Streaming applications consists of six coherent constituents [14]. The block diagram in Figure 2.3 shows the constituents as closely related blocks. The constituents are individually explained ahead.

Video Compression: Raw video must be compressed before transmission to achieve efficiency. Refer Chapter 2.2 for a brief overview of Video Compression. Video compression schemes can be classified into two categories: scalable and non-scalable video coding. Since scalable video is capable of gracefully coping with the bandwidth fluctuations in the Internet, we are primarily concerned with scalable video coding techniques. We will also discuss the requirements imposed by streaming applications on the video encoder and decoder.

Application-layer QoS control: To cope with varying network conditions and different presentation quality requested by the users, various application-layer QoS control techniques have been proposed. The application-layer techniques include congestion control and error control. Their respective functions are as follows. Congestion control is employed to prevent packet loss and reduce delay. Error control, on the other hand, is to improve video presentation quality in the presence of packet loss. Error control mechanisms include forward error correction (FEC), retransmission, error-resilient encoding, and error concealment.

Continuous media distribution services (Internet): In order to provide quality multimedia presentations, adequate network support is crucial. This is because network support can reduce transport delay and packet loss ratio. Built on top of the Internet (IP protocol), continuous media distribution services are able to achieve QoS and efficiency for streaming video/audio over the best-effort Internet. Continuous media distribution services include network filtering, application-level multicast, and content replication.

Streaming servers play a key role in providing streaming services. To offer quality streaming services, streaming servers are required to process multimedia data under timing constraints and support interactive control operations such as pause/resume, fast forward, and fast backward. Furthermore, streaming servers need to retrieve media components in a synchronous fashion. A streaming server typically consists of three subsystems, namely, a communicator (e.g., transport protocols), an operating system, and a storage system.

Media synchronization is a major feature that distinguishes multimedia applications from other traditional data applications. With media synchronization mechanisms, the application at the receiver side can present various media streams in the same way as they were originally captured. An example of media synchronization is that the movements of a speaker's lips match the played-out audio.

Protocols are designed and standardized for communication between clients and streaming servers. Protocols for streaming media provide such services as network addressing, transport, and session control. According to their functionalities, the protocols can be classified into three categories: network-layer protocol such as Internet protocol (IP), transport protocol such as user datagram protocol (UDP), and session control protocol such as real-time streaming protocol (RTSP).

CHAPTER 3

LITERATURE REVIEW AND RESEARCH GAPS

3.1 Literature review

The use of technology in teaching and learning is an active field of research presently. The literature is rife with explorations of what e-Learning is, how it can be used effectively and efficiently, and what role it plays in higher education.

Despite the extensive research in the field of E-Learning, the area of Remote Labs is still quite nascent. Initially people were very doubtful of the educational influences of Remote Labs. A study conducted by Corter et al. [15] gave encouraging results and boosted the implementation of Remote labs at various places. Meile et al. [16] proposed a system for control system laboratory, but it needs 20 fps video transfer and camera for continuous student monitoring, which need very high bandwidth for each and every system for student. And they have used email for communicating orally between instructor and student.

Initial attempts at Remote labs were typical specific solutions to given experiments. For example in [17], a virtual chemical engineering laboratory was developed which was implemented as a supplement to the regular chemical lab course, and it is developed for single course work only but not for general purpose. Bagnasco et al. [18] presented a remote laboratory that models distributed environment characterized by a double client server structure. It consists of a main Virtual Laboratory Server (VLS), a network node hosting a web server that introduces users into the virtual laboratory, implements the access control policy, and logs users' activities and one or more Real Laboratory Servers (RLS).

The remote control of instrumentation is not enough to set up a remote laboratory. Several academic, technical, and structural issues must be faced to obtain generic, modular and scalable systems. Thus, further research targeted practical considerations like accessibility, content, network requirement, multimedia management and streaming etc. In [19], the authors describe a pilot online remote interactive laboratory (RIL) environment used to deliver remote internetworking laboratory experience. They include many aspects like course content, instructional design and multimedia delivery mechanism. The system was tailored to model a synchronous, collaborative, directed learning environment that meet the unique educational challenges of the Internetworking program, supports multiple simultaneous real-time interactions and employs multi-media including streaming video/audio to provide unambiguous equipment cabling and wiring information to multiple concurrent users. But it lacks interactivity with the students. However the current laboratory can accommodate only 30 students maximum in a given time slot and the program requires one remote site facilitator for each remote site. Kikuchi et al. [20] investigated a remote laboratory client-server based framework on electric motors using high-speed networks between Japan and the United States. However, their system requires a network bandwidth of 15 Mbps for efficient remote laboratory experience. Such high bandwidth is generally not possible for majority of E-learning users. The current implementation improves bandwidth utilization by effective image compression and takes very less bandwidth for E-learning users.

Luo et al. [21] and Wang et al. [22] present systems for efficient video transmission over the internet but do not consider the salient features of generic e-Learning videos. This prevents from achieving the optimization levels that are possible. Another system is presented by Mittal et al. [23] and Sood et al. [24] for content adaptive streaming of educational videos. The system employs blackboard and teacher detection and multi-layer encoding for efficient compression. The issue with this system is that it requires a lot of time to encode and decode the video and hence is suitable for offline usage at best. A system proposed by Liu et al. [25] handles these issues and presents a lecture delivery framework for wireless devices. The primary issues with this system are that it is designed primarily for mobile devices and targets only English speaking audience.

Many researchers have addressed the issues related to automatic captioning and annotation. Watanabe et al. [26] had designed Video Caption Markup Language (VCML) (based on XML 1.0) and developed VCML player which can play video data with captioning according to VCML document. Gao et al. [27] introduced the concepts of keywords-based news story indexing and retrieval. Smith et al. [28] have presented the idea of "annotation as argumentation" to help the learners to articulate more than contents summary. Shih et al. [29] proposed multistory annotation system specially designed for distance learning application. In an approach, Wilcox et al. [30] proposed method for indexing and retrieval of multimedia data based on annotation and segmentation. However, the aforementioned works fail to recognize the importance of regional language captioning and the retrieval techniques used by them are solely dependent on English language texts only. But the embedding text media in to video data frames makes harder for the player to search by keyword with in the subtitle and seek to the respective position in the presentation.

3.2 Research Gaps

The literature review revealed following gaps in the current work. The dissertation tries to fill up some of these gaps.

1. Lecture delivery systems of present are not suitable for low bandwidth connections. This is also an issue for end-users who have broadband but their ISPs charge them by amount of bandwidth consumed.

2. Remote laboratory systems are currently targeted to monitoring of hardware equipment or taking measurements via programming logic controllers. Computer-based laboratories haven't been targeted so well.

3. Remote laboratories are yet to simulate the actual Laboratory environment by providing the students an invigilator and by providing the invigilator an evaluation and grading mechanism hence making the remote laboratory as competitive and as exciting as a real laboratory.

4. E-learning systems are yet to incorporate multilingual features. This has prevented far-off reach especially in diverse countries like India.

5. Subtitling systems have not been optimized for streaming video applications. And there is no keywords based efficient seek implemented so far.

CHAPTER 4

BANDWIDTH EFFECTIVE REMOTE LAB MONITORING SYSTEM

4.1 Overview

A system that allows: remote lab monitoring, doubt resolution, synchronous teacher and student participation. The system acts as an alternative to a laboratory session. Instructor can continuously monitor students and Students can interact with the Instructor at any time they want during the laboratory hours. Remote Lab monitoring system can improve quality of education in distance education, by providing practical learning environment for distant users. Remote lab system will reduce the cost of the laboratory infrastructure by providing a way to utilize the students own systems. Rather than sending continuous stream of video from the client screenshot sending can improve the network utilization as well.

4.2 Framework

The proposed Remote Laboratory framework provides complete management and control of labs. The system allows students and instructors to login remotely from their homes or hostel rooms.

The Instructor decides a session (time) to conduct a laboratory, and gives the host information to the students. Students can Login using the remote laboratory system to the Lab Monitoring station. Local instructor of the laboratory will host the Lab monitoring Station at server, so that all students can login.

The Instructor provided with the facility to watch each and every students logging in to the system and what they are practicing on their screens. The instructor can concentrate on a single student and he can suggest that student. He can give group instructions to anybody in the Lab, and he can do it to any individual student. He can warn any student who is misusing the laboratory timings such as, browsing etc.

Students login to the system using their enrolment number as well as their names in the remote lab system, which can be continuously monitored by remote lab monitoring station at server (instructor). Students, given a console to ask any questions to Instructor, if they found any difficulty while undergoing the lab. Students can get continuous group instructions or individual instructions from the instructor at any point of time during the laboratory.

4.3. Architecture

The whole framework can be viewed two parts, Student System and Instructor System. Student System is the part of the application which runs on each student's computer and Instructor System is the other part of the application which runs on the other side, which is instructor or server side of the system (which monitors whole student systems).

4.3.1 Student System

The student application is the software program that the student will start when he wants to participate in the lab. A student will login into the laboratory at the specified time to start the screen capture system. The screen capture system has been written in Java and is platform independent hence not restricting the student to work in a given environment. The screenshots are converted to thumbnails by the system and sent to the instructor's computer (henceforth server) at a rate of 1 thumbnail per 2 seconds. This limit was determined experimentally and it allows the system to consume very low bandwidth and transfer considerably important laboratory events which is one of our primary concerns. The image file format for the process was chosen as JPEG after a subjective comparison [4]. The student system block diagram is given in figure 4.2.

The student can also request a connection with the server (Raise Hand action). This action can be taken either to ask a question from the instructor or to inform the instructor about successful completion of Laboratory assignment. After the teacher accepts the request the student system starts sending full resolution screenshots. This requires high bandwidth on the student side. Hence, a unique strategy has been applied to reduce the bandwidth requirement without hogging up the student's computer system. The new connection has some specific requirements and hence it has some special properties that can be utilized to reduce required bandwidth. A full resolution movie with 30 frames/second is not needed because the rate of change of source is not that high. The user can select the resolution for each screenshot. Since the application is not time critical, the screens are buffered for 5 seconds before being sent.

Each second 2 screenshots are taken so the buffer contains 10 screenshots. algorithm applied to the screenshots is given in figure 4.1

Step 1: An array of 10 adjacent screens named S is created. Two screens are adjacent in the array only when they are temporally adjacent. Timestamps are added to all the screens according to the time the screenshot was taken. These timestamps will be used on the server to display the screenshots in a synchronized manner. The pixel difference between all adjacent frames is computed. D_{ij} is the number of differing pixels between two frames i and j.

Step 2: The *threshold* is set to 1% of the present resolution. Experimentally, this threshold shows good output for text editing as well as graphical changes on screen.

Step 3: While $i = 1$ to $10, j = i + 1$; If $D_{ij} < threshold$, Screen S_i is removed from the array. Hence $S = S - S_i$.

The screens are shifted into the empty slots, hence created. If no empty slots are present, the buffer is transmitted to the server; else, Step 3 is repeated.

Figure 4.1: Buffer algorithm for client

This technique reduces a considerable amount of data transfer without putting overload on either the client or the server. The system at the Instructor side which receives and processes this data is described in the next subsection .

Through interaction system, when ever a student wants to interact with the instructor he can do it. Interaction system is a simple chat room type of interface. Student will receive Public instructions to all students, and private messages from instructor to the respective student. In which, each type of message is mentioned as private or general while displaying to the student. Student want to clear his doubt about some piece of code or some design or any other similar he can send a request to the instructor, and instructor can respond to him. In this scenario instructor can see the screen of the student and clear his problem as well.

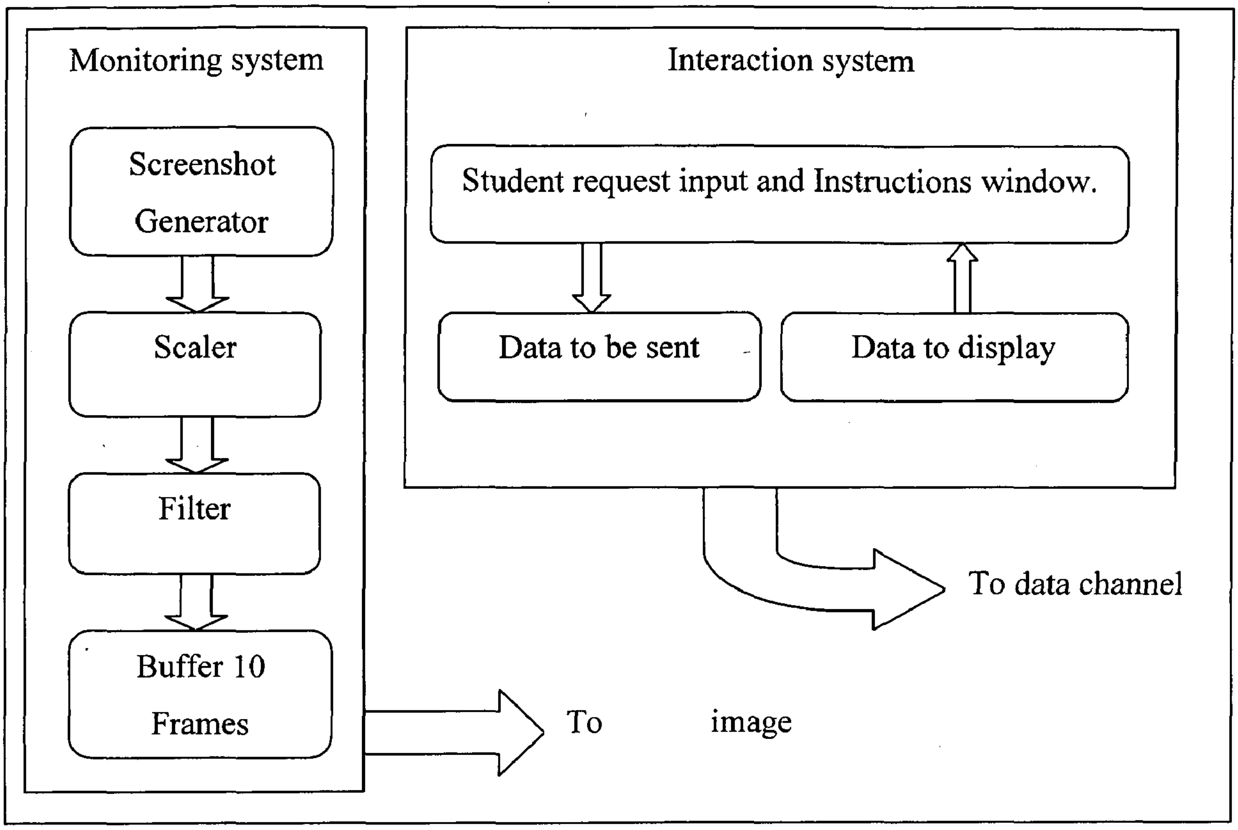

Figure 4.2: Student system

4.3.2 Instructor System

The Instructor's application is more complicated than the student system. It also requires high bandwidth and good processing speed. This was deliberately done to shift the load from the student side to the instructor side. As the laboratory starts, the system shows the thumbnailed windows of students logged into the system as shown in figure 4.3. The thumbnails are identified at connection request and associated with student's enrolment numbers. The instructor simultaneously monitors all the students' screens. The minimum bandwidth required by the instructor system for this purpose is the product of the number of students logged in and the bandwidth required for transmitting each thumbnail.

Applying the reverse decoding process in that case requires extremely high processing power from the workstation and slows it down considerably. The interaction system is a Chat system, in which the server will initiate at first, so that all the client (students) systems can join the server. And there will be facility in server such that Instructor can select the student out of the students who are connected and can have a private conversation with him. Otherwise he can directly broadcast the message to all other students in the Laboratory session. Whenever Instructor finds to suggest or warn a student he can see the name in the thumbnail (display) window and he can select that particular student in the Interaction window and can have a private conversation with him, by double clicking on his name, which will lead to a dialogue box, where he can type the message.

The Interactions system works in such a way that, it is a server chat thread which will wait for any client to join to the server, and authenticates them and joins them to the server,

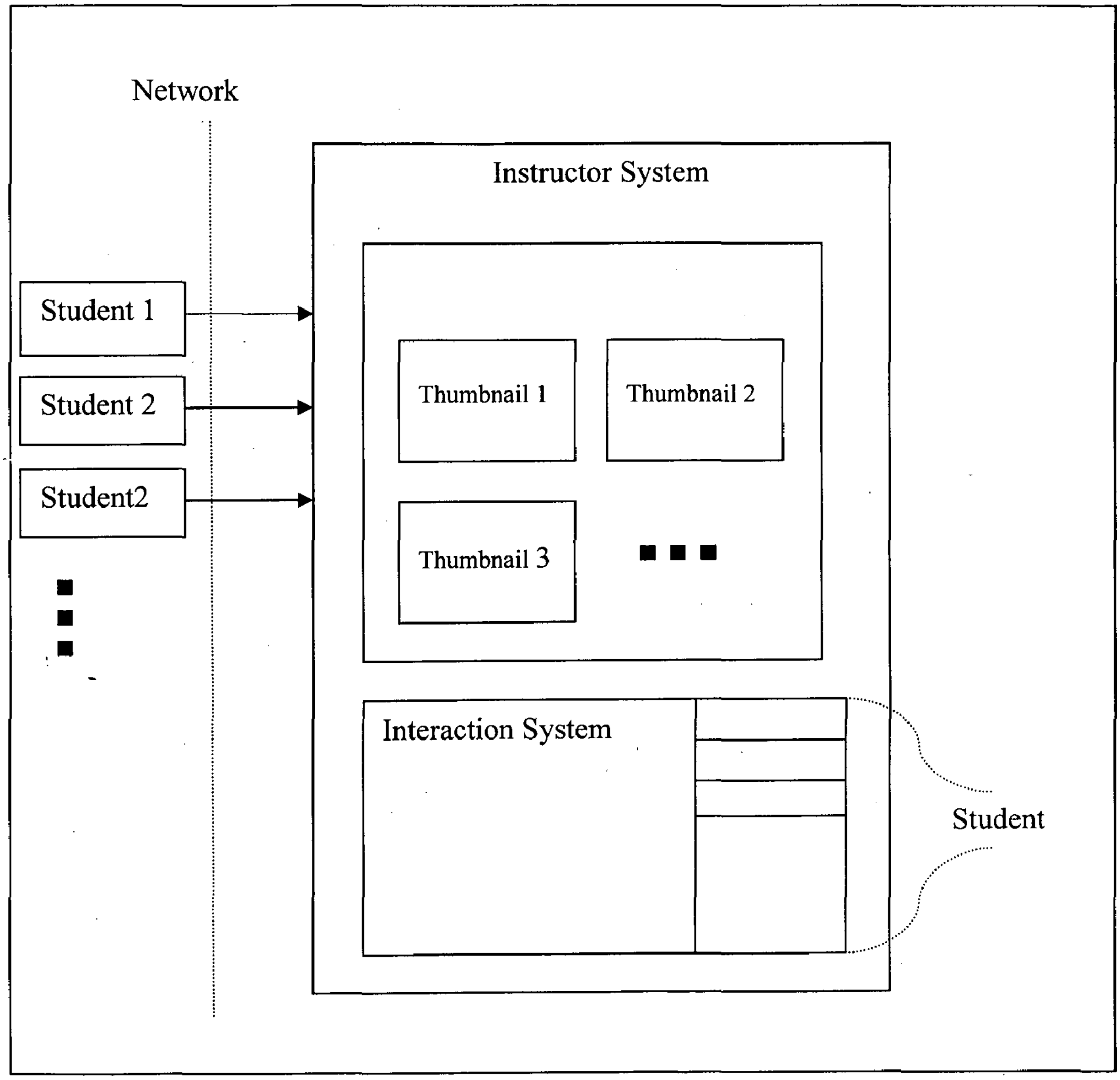

Figure 4.3: Remote Lab Monitoring system block diagram.

which will work parallel with the screenshots server. So, each and every student whose screen is being captured will be entitled in the interaction window such that the Instructor can communicate.

Once the instructor selects a thumbnail and opens up the new screen the system starts receiving the detailed screen buffer from the student system (henceforth client). This is as shown in the sample screenshot. The buffer initially had 10 screens as it waits for 5 seconds with 2 screens per second but the buffer received after applying the scheme may contain from 1 to 10 screens. These screens contain timestamps that were added at the client side. Algorithm used to display the screens in a time synchronized manner is given in figure 4.4.

Step 1: The received screenshot buffer has already been sorted at the client side so the first thumbnail, the one with the lowest timestamp, is displayed.

Step 2: The difference between timestamps of two adjacent screens is calculated. $t_d = t_{i+1} - t_i$, where t_i is the timestamp of frame i and t_d is the difference in timestamps.

Step 3: The system waits for time interval t_d and displays the next screenshot.

Step 4: The process is repeated till the buffer is empty

Figure 4.4: Algorithm for buffer management at server

The instructor can also establish a connection with the student on the detailed monitoring screen. This can be utilized either to answer a raise-hand request or to ask a question as part of the laboratory grading process. For e.g. in a programming oriented laboratory the instructor can ask the student to explain the particular code he has just written and grade him accordingly. The assignments can be reviewed by the teacher remotely just like real laboratories. The students can also get their assignments checked remotely just like they are accustomed to do in real labs.

4.4 Implementation

4.4.1 Languages and Framework

The Remote Laboratory system works as a protocol designed over the basic client server model. Seeing it's application (laboratories can have computers with various environments), it was ported to Java using JDK 6 enhancing its platform independence. For, the basic image processing functions and algorithms Java Imaging API was used. And the IDE used for this is

Netbeans 5.5.The window interfaces are implemented from java swing. Java net package is used for network programming in the project, like creating sockets and data transfer (images, text).

4.4.2 Implemented Package List

Client side: Default package contains Client class which contains main function for the client program. In this the program will divide into two threads, one is main thread and another is chat thread. Main thread will continuously monitor the client, which means takes the screens shots and send the thumbnails or full images as required. The other thread will initiate the chat window. Screenshot class creates screenshots and converts to desirable format according to the request. Request class will serve each request by creating socket and sending the image file to the server. Newthread class is a thread which implements thread and invokes p2pchat awt frame. A screenshot of client is given in Figure 4.5.

Another package is p2pchat which will contain whole bunch of classes which are responsible for generating the chat frame and handling exceptions and creating client to server connection for interaction etc. p2pchatAWT is the class which contains the main function for this thread, which is invoked by a separate thread in the main function at client class. P2pchatAWT is implemented from abstractp2pchat class, which handles all the chat window sockets and connections. And it also has the code for the Interface window for the client application.

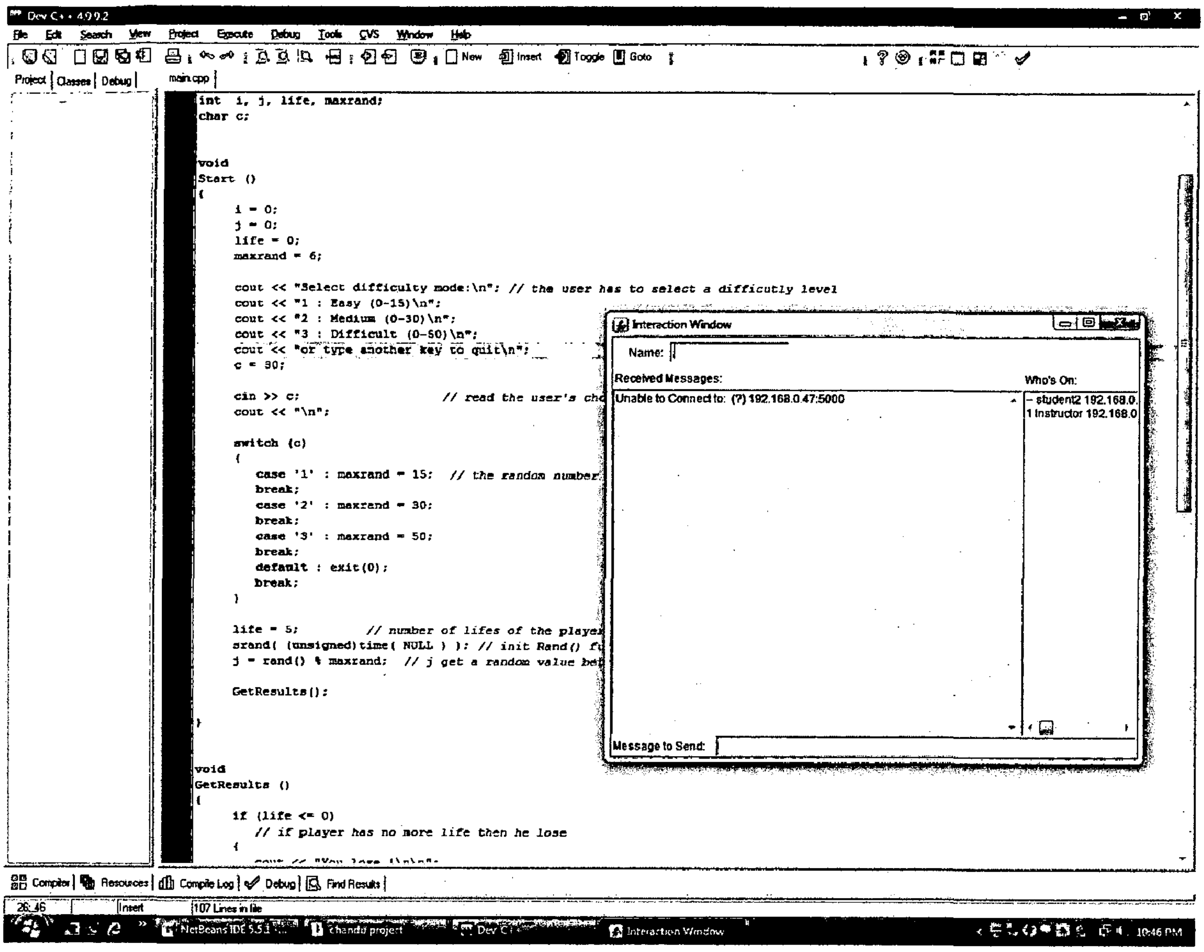

Figure 4.5: Student working in lab

Server side: Other than p2pchat package which is explained above, there is another package which is important for server side: remotelabapplication package. This package contains **Server** class which is the class for server socket and accepts student's socket connections here. Request class will fulfill the pending requests which are generated by the each client socket for each screenshot. Vlauepair class is simple data structure to store the information about each client. **Screens** class is main class, which invokes newthread class as another thread for Interactive (chat) window. And main thread will give thumbnail display of each client. **Screenshot** class will display individual enlarged screenshots of students. A server screenshot is given in figure 4.6.

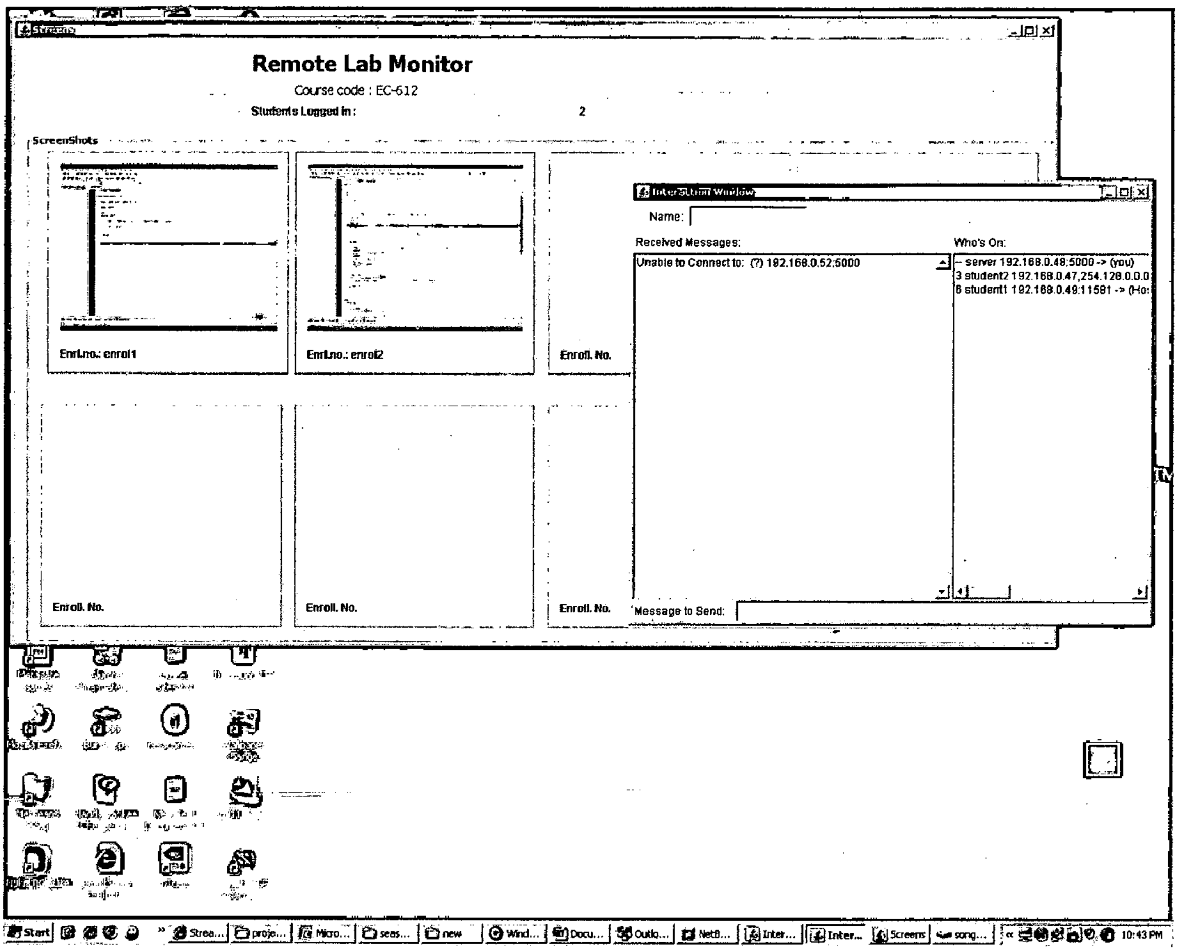

Figure 4.6: Server with students connected

4.4.3 Problems faced

Scalability: The Remote Lab system is not highly scalable in the sense that it was designed to work for labs with utmost 30 students. Adding students beyond will render it unusable in two ways. One, the bandwidth and processing requirement at server side will be extremely high. Two, it will over ride the maximum limit of screen that the server application can monitor. Even if the limit is updated the instructor will not be able to simultaneously monitor over 30 students on a single screen, however large it may be.

Efficiency: The Remote Lab system is highly efficient in all the paradigms, be it time, space or bandwidth required. The client application takes screenshots and stores them directly as resized images. For greater efficiency and quality jpeg format was chosen for this application. On the server side the load is higher but it still only has to display multiple images, just like a web browser which is not very unreasonable in present circumstances.

Failures and Successes: The Remote lab system works fine in most of the cases. But, few cases of failure arise as discusses in scalability issues. Even below the maximum limit the load on instructor is quite high and it's better to have multiple instructors. In a way, it is analogous to real laboratories where it's difficult for a single instructor to manage a lab with large number of students.

CHAPTER 5

ADAPTIVE LECTURE VIDEO DELEVERY SYSTEM

5.1 Overview

A lecture delivery system is a primary component of an e-Learning implementation. Currently, the most popular commercial video delivery solutions include Apple QuickTime, Microsoft Windows Media and RealNetworks RealVideo. While QuickTime and RealVideo support a variety of platforms (e.g. Solaris, Linux and Windows), the Windows Media server only runs on Windows and the client only supports Windows and MacOS.

One fundamental problem with all these commercial solutions is that they are optimized for high bandwidth connections: 500kbps for RealVideo, 700kbps for QuickTime, and 720kbps for Windows Media [31]. However, a useful solution for students is one that minimizes the client side equipment overhead requirements. The problem is present not because the solutions are sub-optimal but because they are not customized for e-Learning videos. They have not handled the special conditions that are present in e-Learning videos that can be effectively used to reduce the size even further. The goal of the proposed framework is to provide students with a bandwidth efficient framework.

5.2 Framework

The framework of compressing and streaming instructional videos is shown in Figure 5.1. The incoming video stream is first processed by a "content analysis" module which segments content regions from irrelevant regions for each frame and derives a heuristic measurement of content. A "frame rate control" module then selects content significant key frames based on control parameter – the compression ratio . Another module, "image quality control", changes image quality of output frames by locating content regions, resampling images, and using different color depths. Framework is based on Exploring Multimedia Transmission Issues over Wireless Channels and their Application in Telemedicine [32].

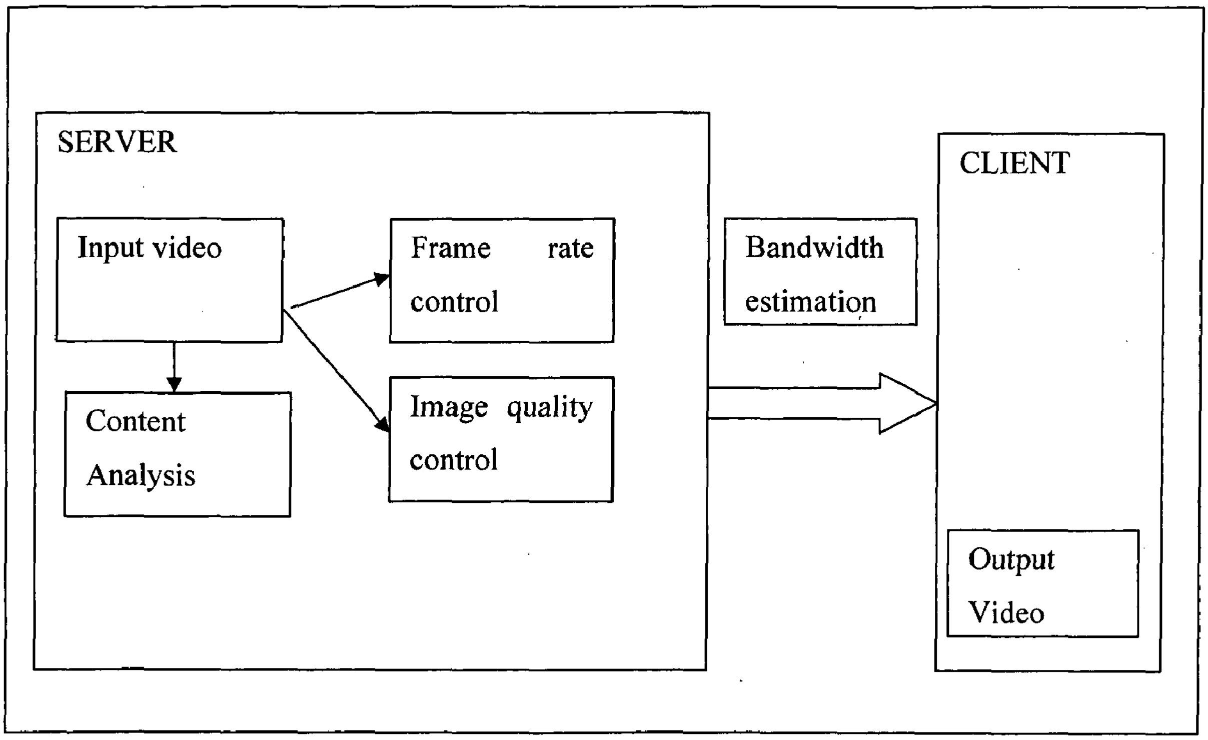

Figure 5.1: Block Diagram of the System

5.2.1 Content analysis:

We identify that every frame of a e-learning video contains an area which is of interest at a given time to the viewer and which we call the region of interest (ROI). The remaining part has little or no information of interest or may contain redundant information. We call this the region of neglect (RON). It is clear that high quality is desired only in the ROI while the RON may get away with an inferior quality. Thus in order to decrease the size of video we will a higher degree of compression in the RON while the ROI is kept compression free. The point of contention is how we can efficiently decide the ROI in a given situation. As a solution to this problem we calculate the similarity of each frame with the template frame by calculating the mutual information between the two. If the similarity drops below the threshold value, a parameter that can be tuned empirically, we know for sure that a mismatch has occurred and thus apply registration operation to the current frame in order to align it with the template frame, so that the desired ROI can be mapped on to the actual ROI in the current frame. From figure 5.2 and 5.3 we can see clear ROI.

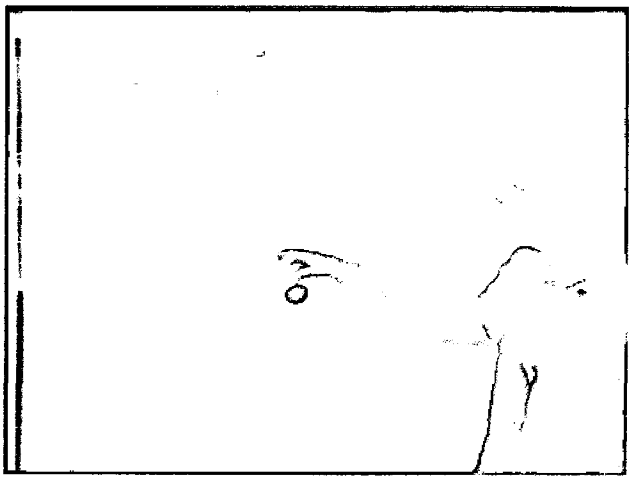

Figure 5.2: Content frame with text on blackboard.

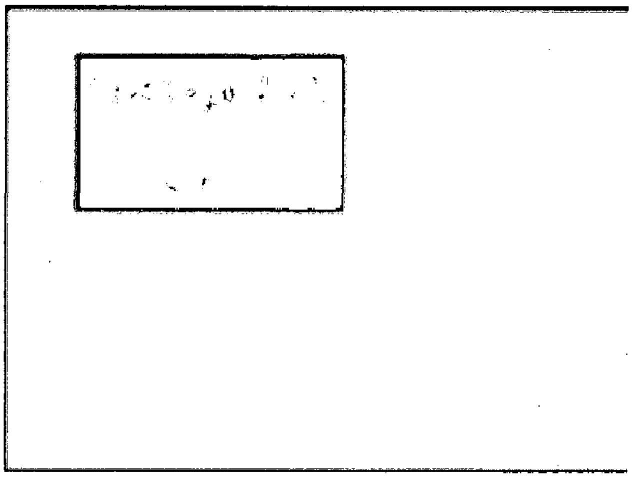

Figure 5.3: Content frame with after Region of Interest.

If the similarity drops below a critical value, another empirical parameter, we know that the frames are too dissimilar to be aligned, and the doctor is sent another uncompressed frame and is prompted for a new ROI. This scheme works well for relatively slowly varying videos.

The overall system is depicted in the figure 5.1, at a functional block diagram level. Thus the system contains a skeleton of functionalities relating to the streaming of videos between a client and a server. However the innovativeness of the systems stems from a set of modules, which are described below, that give the actual improvement in the performance metrics of the streaming of the domain specific videos.

In **region based compression module** different portions of the image are compressed with different amount of degrees of compression. Thus for example a specified region may be left uncompressed, whereas the remaining regions will be encoded with a lossy mode of compression. The degree of compression depends upon the amount of tolerance we can have. The problem with an independent implementation of two different modes of compression in different areas of the image lies in the fact that images are generally compressed by using the spatial redundancies. Thus a compressed image contains delocalized information about various regions. In order to achieve such a compression we must treat the differently encoded images as being independent of each other this implies while transmitting the image we must send these two images together. For example if the doctor specifies his region of interest we must code the region one image and the remaining region as separate image. We then package the two images in a single payload and transfer it. This approach incurs some inefficiency due to sacrificing the spatial redundancies to an extent. However as our results show, the overall gain in terms of reduction of size with respect to the original uncompressed images outweighs this small inefficiency.

5.2.2 Frame rate control

The output video frame rate λ depends not only on bandwidth, but also on video content. For the content frames the frame rate is kept lower since a high frame rate is simply not required. This part of video only shows text being updated and updating text at 30 frames/second along with the varying frame size (in some cases) can actually be confusing and illegible in case of blackboard or handwritten projection. In the case of presentation slides even 1 frame/second is sufficient as generally each slide stays on screen for more than 1 second. The frame rate is limited at minimum 1 and maximum 5 frames per second. For the case of non-content frames the frame rate varies from 15 frames/second (minimal frame rate before video playback becomes choppy) to 30 frames/second (maximum perceivable frame rate for human eyes). But non-content frames are not sent to this module because they are handled by the video compression module that allows setting the frame rate as pre the requirements.

For adjusting the frame rate the "leaking video buffer" proposed by Liu et al. [25] is used.

Leaking Video buffer is inspired by human short-term memory recognition process and the flow control in computer networks, a "leaking buffer" model is provided to select key frames dynamically. The "leaking video buffer" at the server side consists of n slots, each of which holds one video frame. A video sequence comes in at one end of the buffer at a frame rate of λ_{in} and leaves the buffer at the other end. The output frame rate, λ, is determined by the video content and the network condition. An evaluating process is applied to evaluate all the frames in the buffer and the comparatively insignificant frames are "leaked" from the buffer. The output key frames are those "pulled out" at the other end of video buffer. The frame leaking decision is made each time when there is no empty slot in the buffer.

5.2.3 Image quality control

This module estimates the current bandwidth available for the streaming of the video. We use a method similar to the single packet bandwidth estimation. The client side estimates the bandwidth available as the mean of the time taken to receive last 5 packets. If however the time taken to receive a packet falls below a critical value we identify it as a critical situation. Depending on the bandwidth estimate the client sends a control packet to the server.

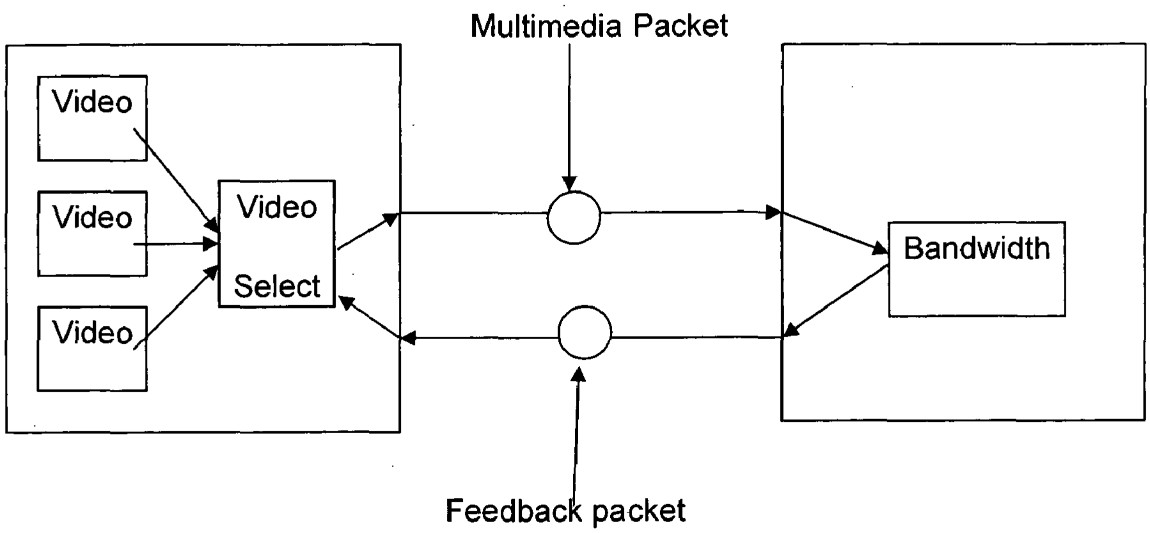

Figure 5.4: Bandwidth Adaptive Video Selection

The server has a set of three videos, each with different degree of compression in the RON. The server then chooses from one of these videos, sending a more compressed video for a lower bandwidth estimate and vice versa. Figure 5.4 shows the selection of videos.

5.3 Implementation:

The following steps were involved:

Step 1: Using JAVA streaming libraries to implement MJPEG streaming module

Step 2: Using JAVA image processing libraries to implement the compression, decompression and image registration module in JAVA

Step 3: Interleaving the sound data along with video data to be streamed over the network

Step 4: Integrating all the modules together

Interaction between different modules in the system is given by figure 5.5

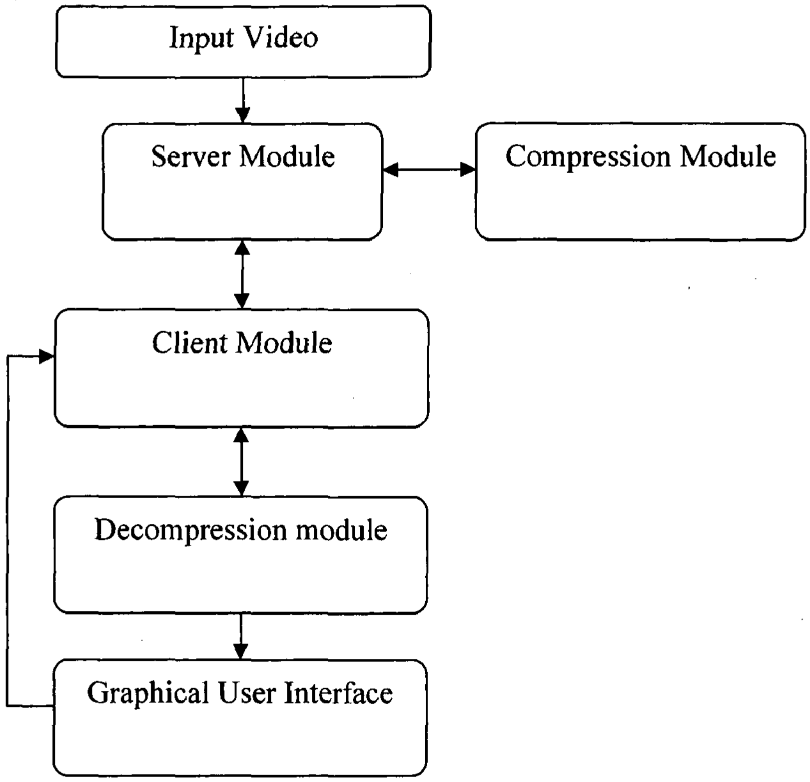

Figure 5.5: Different Development modules in the system

CHAPTER 6

A MULTILANGUAGE CONTENT MANAGEMENT SYSTEM

6.1 Overview

A Multilanguage content management system for e-learning content (video, audio, text, clips etc). For this we described the design of Multilanguage content markup language (MCML) and development of a player for it. In MCML we can describe a content time of presence in the presentation by providing its time and position information. And it also has Multilanguage subtitle provision. And through MCML player can seeks video/presentation to a specific time by searching by content description or subtitle.

Existing multimedia software in E-Learning does not provide adequate excellence multimedia data service to the common user. Hence E-learning services are still short of intelligence and sophisticated end user tools for visualization and retrieval. An efficient approach to achieve the tasks such as regional language captioning system and keywords based efficient seek, is introduced.

6.2. Authoring System

This section introduces the overview of the video, image data, speech data, and text data flow diagram shown in Figure 6.1. The proposed system receives one or more inputs some are video data, text sequence, audio/spoken by speakers.

MCML Authoring: Video are lecture video files corresponding to a specific lecture topic. Authored MCML document will contain information of from which time video starts to play to which time at which position, as well as other media items. The text can be of two different languages, specified with what language it is. The text generally we can call it as subtitles. These subtitles are also containing timing information for synchronization with other media types.

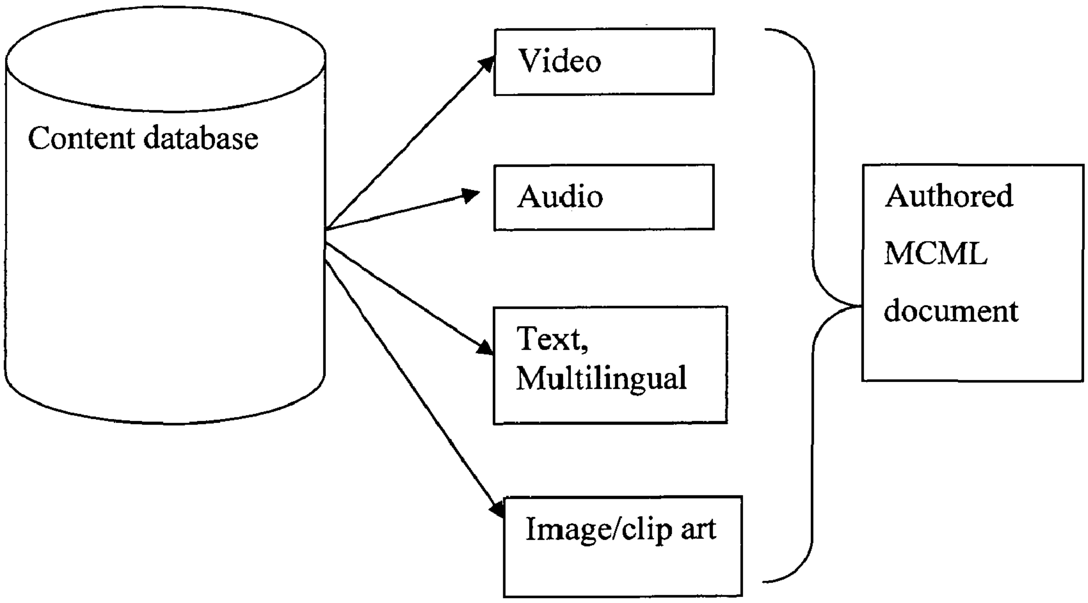

Figure 6.1: MCML Authoring system

6.3 Design

XML (Extensible Markup Language) **as** a markup language for future web browser, is selected for designing a MCML.MCML is a XML document which consists of three parts, XML declaration, DTD(Document Type Definition) and XML instance. XML declaration defines character set and XML version. DTD defines usage of tags, attribute list and the list is necessary or not. XML instance means actual document [33].

6.3.1 MCML Tags

Document head: In the head element of the MCML document, **<title>, <author>, <date>, <copyright>** and **<abstract>** tags set title, author, date, and copyright of the document, respectively. **<rootlayout/>** set the screen size of displaying area and **<region/>** defines the structure of the screen area and names it.

Rather than any other multimedia presentation languages, we will divide the content of the MCML Document in intervals. These intervals we can call as Topic. So finally, A MCML document called as a title and the <topic>s are intermediate discussions in that title. This topics are further divided based on content called <audio>, <video>, <slide> and <subtitle>. A topic can have multiple these content tags and these content characteristics are like "timefrom", "timeto" and region (layout). Unlike other content, <subtitle> content will have language and font characteristics. Font is same as HTML font.

6.3.2 MCML Player

An XML parser reads MCML document, then, the parser makes a syntax analysis. Through the parsed information MCML player generates event chain for the presentation. And each MCML document is presented topic by topic. For an instance a video with subtitle can be followed by an image with audio etc. Block diagram of the MCML Player is shown in Figure 6.2. Time Controller counts the process time from the beginning. "PLAY" command reset the process time of Time controller as 0, and event entry part and event execution part start. Event entry part searches event table, and dl events whose beginning time is greater than the process time, is added to active event lit. Next searching starts from the next event of the previous executed one. Time controller checks active event list, and if the ending time is smaller than the process time, the controller deletes m event from the list.

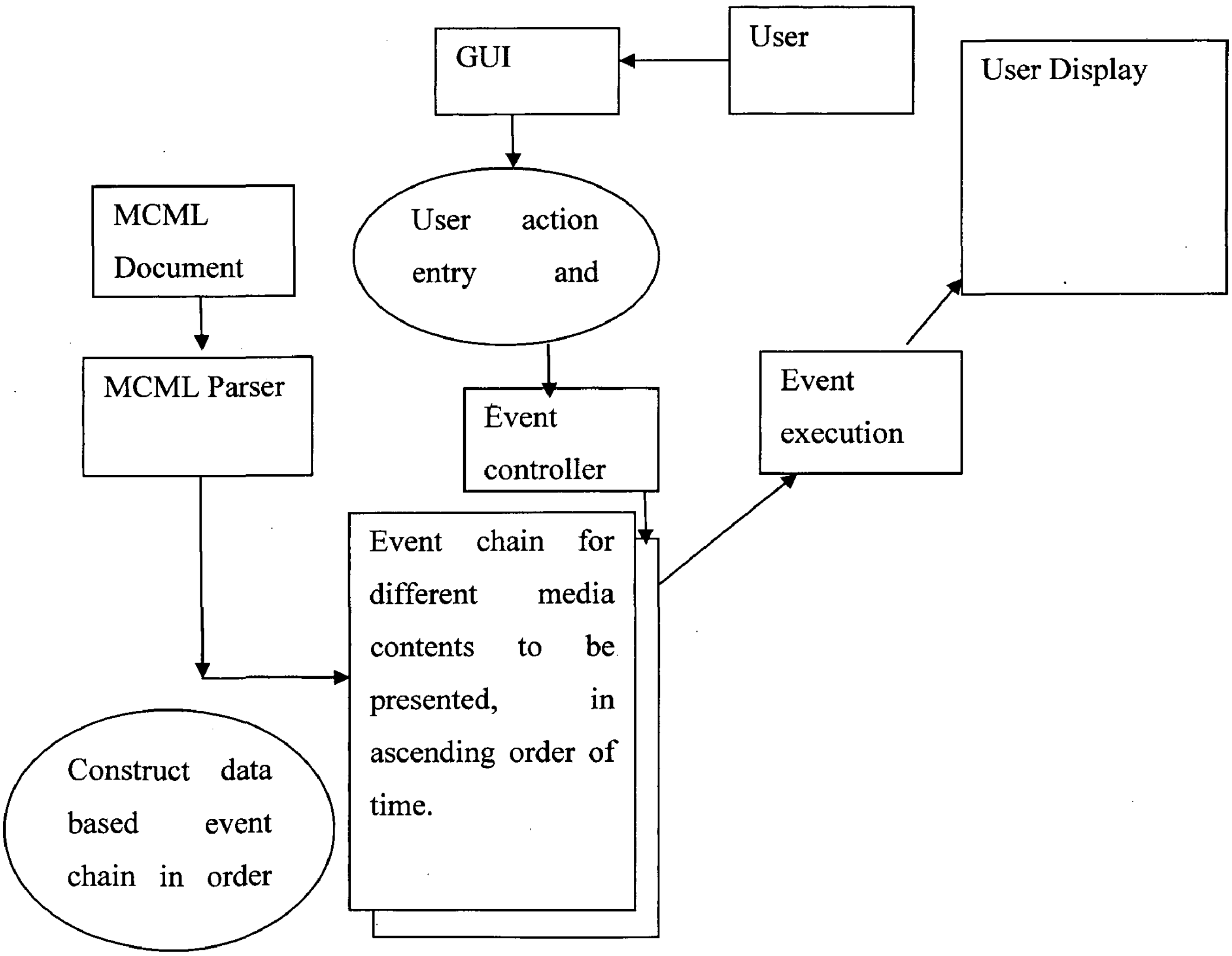

Figure 6.2: MCML player

Event execution part displays event data in active event list. In between user can search for a string in subtitle or topic in the whole title, when this event occurs player directly seek to the topic of search if the searched string topic, other wise it will give possible matches in subtitle for the specified string. And the jumps to that specific position and play the presentation from there.

6.4 Implementation

The basic development tools and libraries used for the development of the MCML player is Microsoft® Visual Basic 2008. The MCML content supplied to MCML player is parsed by XML parser, and the event chain is generated. And this event chain output is given to MCML player User Interface application window.

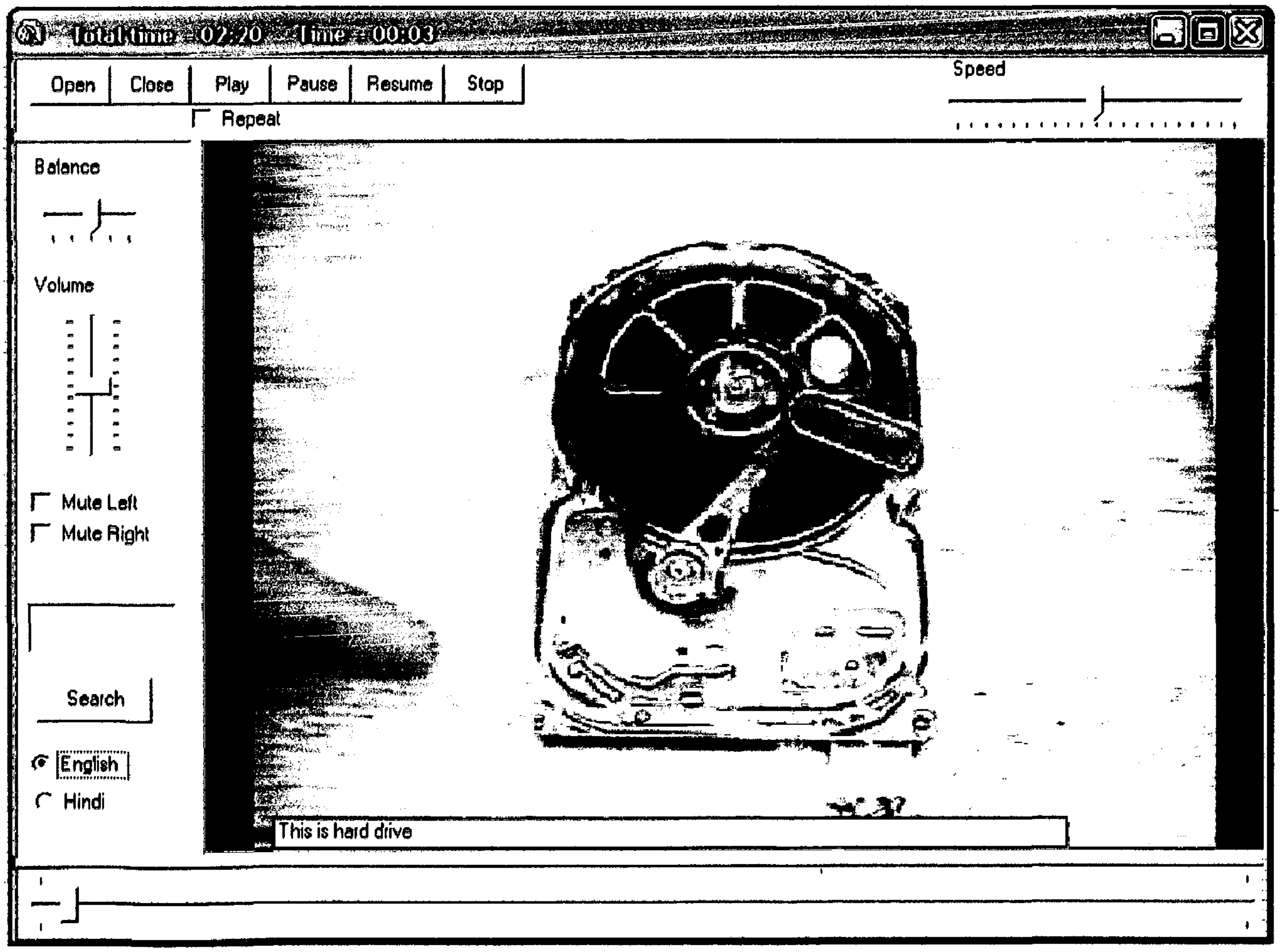

Figure 6.3: Screenshot of MCML player

Implementation started from creating DTD (Document Type Definition) for MCML in XML. And MCML document is written in Unicode16 rather than ascii to support multiple language authoring.

Figure 6.3 shows working screen shot of the MCML Player, in which English subtitles are being despalyed.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Remote Laboratory system

The remote lab system was tested with various image formats to choose the most suitable one. Figure 7.1 shows a comparison the bandwidth required by basic monitoring system at student side. The initial subjective conclusion was that GIF file format worked best for our system. But once the system used, it was discovered that the images received were very bad especially in case of modeling applications as GIF resulted is heavy color approximations and hence loss in quality. Finally JPG was selected for thumbnail monitoring purpose in the system.

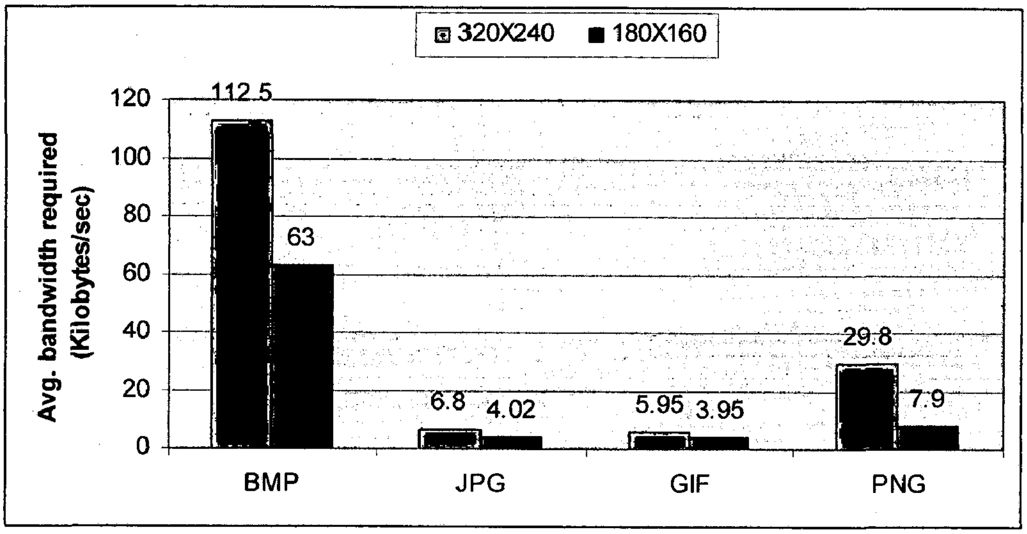

Figure 7.1: Comparison of bandwidth required by basic images formats and sizes.

The tests for average buffer sizes for transmitting the detailed student screen were conducted on a number of cases to yield interesting results. Figure 7.2 and 7.3 provide a comparison of average buffer sizes for transmitting the detailed student screen for the cases of a generic C++ programming Laboratory and a Computer-aided Design lab respectively. The difference in requirements of bandwidth is attributed to difference in laboratory setup. The relative performance of image formats also varied. We thus selected both JPEG and PNG for transmitting the detailed student screen. The screenshots in the both the cases were taken with the resolution set to 1024x768.

From the results, it is clear that for different applications different image formats provide best results. GIF provided the best results in both cases but the quality of images was simply unacceptable. Hence, an option was added to the system for transmitting the thumbnails in PNG or JPG.

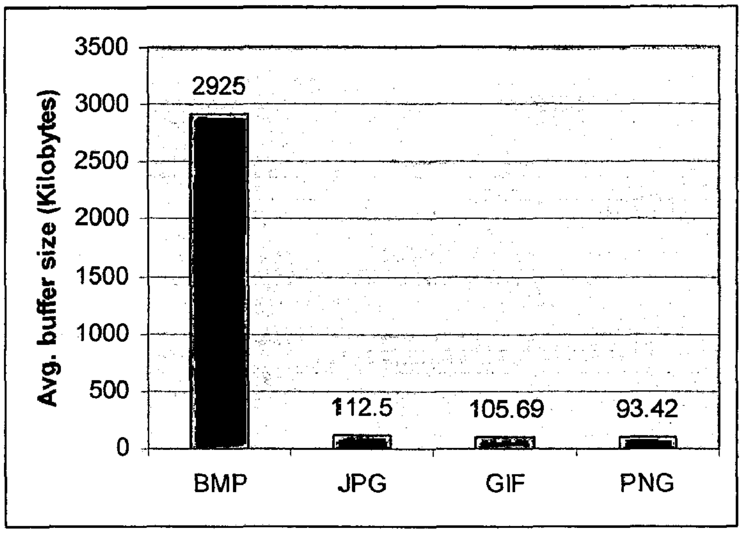

Figure 7.2: Comparison of Average buffer size in a Programming Laboratory while transmitting detailed student screen using various image formats.

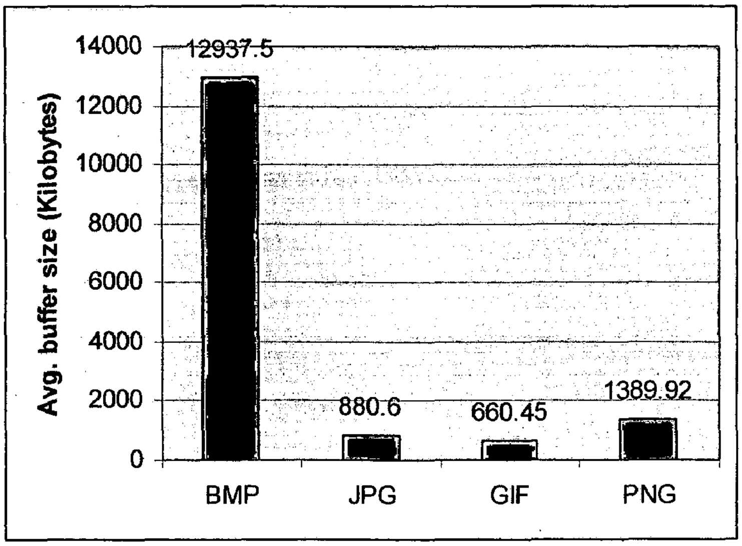

Figure 7.3: Comparison of Average buffer size in a Computer-Aided Design Laboratory while transmitting detailed student screen using various image formats.

This allows the students and instructor to select the optimal format beforehand. The choice of proper image formats makes the system more efficient and a practical solution to implement remote laboratories through the internet.

7.2 Multilanguage content management System

Multilanguage content management system can compete with existing markup language for content captioning; the following table describes the Advantages over the already existing system.

Feature supported	Proposed method	VCML
1.Keyword specific video seek	Yes	No
2. Chaptering a video	Yes	Possible
3. Modifiable subtitle orientation during playback	Yes	No
4. Authors control over different media types in presentation	Yes	Not
5.Supports automated multi-lingual audio generation	Yes	Not possible
6. changing between multiple languages ant runtime	Yes	No
7. Search in multiple languages	Yes	No
8. Regional language support	Yes	Yes

Table 7.1: Advantages of proposed method over existing method

7.3 Adaptive Lecture video delivery system

The lecture delivery system reduces considerable bandwidth as compared to streaming a compressed video. In the given case, the input video is a 640 x 480 M-JPEG video taken using a standard camera. For comparison, the video is encoded to fit a 128 Kilobytes/sec stream. The plot in Figure 7.4 shows the performance benefits of the lecture delivery framework as compared to normal streaming systems.

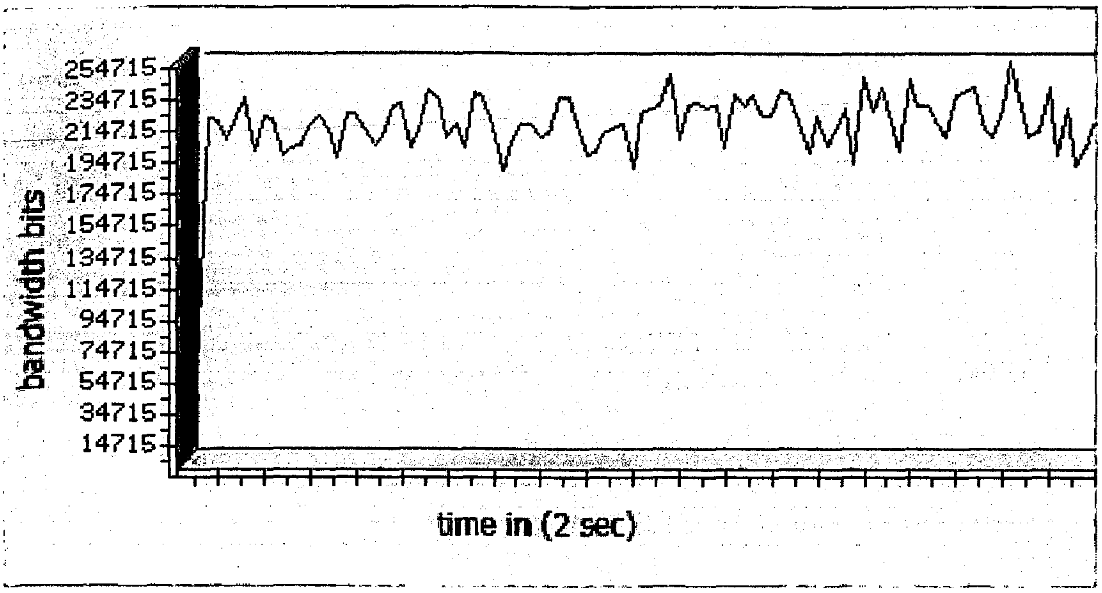

Figure 7.4: Data streamed by lecture delivery framework server, blue line shows the standard video stream at constant bit-rate of 128 Kilobytes/sec.

The plot shows data streamed which for the given system is precisely equal to the bandwidth estimation given by the network traffic monitor tool. The plot demonstrates that the system works much better than normal video streaming systems for lecture videos with enough content frames.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In this dissertation we presented multimedia delivery solutions for e-Learning comprising of multiple frameworks and a system has been proposed. The system comprises of three parts, a framework for efficient delivery of lecture videos, a Remote laboratory monitoring system and a Multilanguage content management System. The system if properly adopted, customized and utilized, can be a great solution to the problem of illiteracy faced by India and several other developing nations. It provides answers to many of the major issues hampering wide usage of E-Learning systems including bandwidth constraints and language barriers.

The Remote Laboratory system focuses on simulating a real life competitive Laboratory environment as closely as possible. The framework provides instructor friendly remote monitoring of lab, effective evaluation and grading methodology. The system also provides student friendly remote login, software access and problem resolution through effective help from the teacher. The system is highly bandwidth efficient and allows students to participate in labs even from far-off places with no broadband connectivity. This fact has been proven via the results in Chapter 7.1. Although the pedagogic abilities of the system have not yet been tested, it presents great new possibilities in the area of e-Learning.

The lecture delivery framework proposed in the dissertation provides better solution to streaming videos than current generic solutions. The system allows dynamic content analysis and application of quality control and strategic compression algorithms to reduce the required bandwidth. This fact has been affirmed by the results in Chapter 7.3. The web-based streaming system is a novel idea for e-Learning lecture videos. The system has immense potential in terms of future enhancements and integration with the other systems presented in this dissertation.

The Multilanguage content management System bridges the language gap in lecture video based e-Learning. It allows people with hearing impairments to watch and understand various educational videos. The project can also be used to provide vocational training to illiterate people in different villages. The content based search method in player is very helpful for interactive navigation in a presentation. The system is highly efficient as discussed in

Chapter 7.2. It can also be used with other e-Learning systems allowing multilingual features in those systems.

8.2 Future Work

The presented solution can be enhanced in various ways as discussed ahead.

1. A lab examination system can be devised to allow the Remote laboratory system handle a complete lab course.
2. The content analysis step can be improved by using a board detection algorithm on the video. This can simplify the text detection process and further reduce errors in content analysis step.
3. Rather than on stored media lecture delivery system can be improved to work on live video capture at a class room.
4. An automated authoring tool can be developed to improve the usage MCML format.
5. The DTD of MCML can be published for general purpose content management as well for different presentation contents other than e-learning.

REFERENCES

[1] E-learning, *http://en.wikipedia.org/wiki/E-learning*, (last accessed 28th June, 2008).

[2] M. H.Fallah, W. J. How and R. Ubell, "*Blind scores in a graduate test: Conventional compared with Web-based outcomes*", ALN Magazine, vol. 4, no. 2, 2000.

[3] S.R. Hiltz and M.Turoff, "*What makes learning networks effective?*", Communications of the ACM, Vol. 45, no. 4, pages 56-59, 2002.

[4] L. Ze-Nian and M. S. Drew, "*Fundamentals of Multimedia*", Pearson education, First edition, 2004.

[5] H. Ahmad, Z. M. Udin, and R. Z. Yusoff, "*Integrated process design for E-Learning: a case study*", The Sixth International Conference on Computer Supported Cooperative Work in Design, pp. 488 – 491, 2001.

[6] P. R. Polsani, "E-*Learning and the Status of Knowledge in the Information Age*", International Conference on Computers in Education, pp.952 – 956, 2002.

[7] K. Seki, W. Tsukahara, and T. Okamoto, "*System Development and Practice of E-Learning in Graduate School*", Fifth IEEE International Conference on Advanced Learning Technologies, ICALT, pp. 740 – 744, 2005.

[8] S. M. Jeong and K.S. Song, "The *Community-Based Intelligent e- Learning System*", Fifth IEEE International Conference on Advanced Learning Technologies, ICALT, pp. 769 – 771, 2005.

[9] Languages and Dialects in India, *http://en.wikipedia.org/wiki/India,* (last accessed 28th june, 2008).

[10] S. Saha, *"Image Compression - from DCT to Wavelets: A Review"*, Crossroads – Data Compression, The ACM Student Magazine, vol.6, no.3, pp. 12 - 21, 2000.

[11] D. Salomon, "*Data Compression: The Complete Reference*", Springer, Fourth edition, 2004.

[12] R. Prasad, C. Dovrolis, M. Murray and K. Claffy, *"Bandwidth estimation: metrics, measurement techniques, and tools"*, IEEE Network Magazine, vol 17, no.6, pp. 27-35, 2003.

[13] *Real Time Protocol*, available at http://icapeople.epfl.ch/thiran/CoursED/RTP.pdf (last accessed on 28th June 2008)

[14] D. Wu, Y. T. Hou, W. Zhu, Y. Q. Zhang and J. M. Peha, *"Streaming video over the Internet: approaches and directions"*, IEEE Transactions on Circuits and Systems for Video Technology, pp. 282-300, vol. 11, no. 3, 2001.

[15] J. E. Corter, J.V. Nickerson , S. K. Esche, C. Chassapis, *"Remote versus hands-on labs: a comparative study"*, 34th Annual Frontiers in Education, vol. 2, pp 17-21, 2004.

[16] A. M. Deborah, P. Benjamin, T. W. John, *"An Internet-based Remote Laboratory for Control Education"* Proceedings of the American Control Conference Arlington,vol. 2, pp. 1151-1152, 2001.

[17] R. C. Nippert, *"Using Web Based Supplemental Instruction for Chemical Engineering Laboratories"*, Proceedings of the American Society for Engineering Education Annual Conference and Exposition, USA, 2001.

[18] A. Bagnasco, A. M. Scapolla, *"A Versatile Remote Laboratory"*, Computer Aided Learning in Engineering Education, CALIE04, Grenoble, France, 2004.

[19] S. C. Sivakumar, W. Robertson, M. Artimy, N. Aslamg, *"A web-based remote interactive laboratory for Internetworking education,"* IEEE Transactions on Education, vol. 48, no. 4, pp. 586- 598, 2005

[20] T. Kikuchi, S. Fukuda, A. Fukuzaki, K. Nagaoka, K. Tanaka , T. Kenjo, D. A. Harris, *"DVTS based remote laboratory across the Pacific over the gigabit network"*, IEEE Transactions on Education, vol. 47, no.1, pp. 26 – 32, 2004.

[21] J. Luo, I. Ahmad, Y. Liang and Y. Sun, *"Motion Estimation for Content Adaptive Video Compression"*, IEEE International Conference on Multimedia and Expo, ICME 2004, vol. 1, pp. 1427-1430, 2004.

[22] J. R. Wang and N. Parmeswaran, *"Intelligent Streaming Video Data over the Web"*, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, WI 2004, pp. 20-24, 2004.

[23] A. Mittal, S. Gupta, S. Jain and A. Jain, *"Content-based adaptive compression of educational videos using phase correlation techniques"*, Springer Special Issue on Multimedia Systems, vol. 11, no. 3, pp. 249-250, 2006.

[24] A. Sood, D. Sarthi, A. Pande and A. Mittal, "*A Novel Rate-Scalable Multimedia Service for E-Learning Videos using Content Based Wavelet Compression*", Annual India Conference, pp.1-6, 2006.

[25] T. Liu and C. Choudary, "*Content-adaptive wireless streaming of instructional videos*", Springer Multimedia Tools and Applications, vol. 28, no. 1, pp. 157-171, 2006.

[26] K. Watanabe, N. Fukada and M. Sugiyama, "*Design of Video Caption Markup Language VCML and development of VCML player,*" IEEE conference on Multimedia and Expo (ICME), vol. 1, pp.163-166, 2000.

[27] G. Xinbo, X. Hong and J. Hongbing, "*A study of intelligent video indexing system,*" Proceedings of the 4th World Congress on Intelligent Control and Automation. vol. 3, pp. 2122-2126, 2002.

[28] B. K. Smith, E. Blankinship and T. Lackner, "*Annotation and education,*" IEEE Multimedia, vol. 7, no. 2, pp. 84 – 89, 2000.

[29] T. K. Shih, Y. C. Liao, H. B. Chang and M. Y. Kuan, and G. Yee, "*Multistory annotation system: a novel application of distance learning,*" 18th International Conference on Advanced Information Networking and Applications (AINA). vol. 2, pp. 116 – 119, 2004.

[30] L. Wilcox and J. Boreczky, "*Annotation and segmentation for multimedia indexing and retrieval,*" Proceedings of the Thirty-First Hawaii International Conference on System Sciences, vol. 2, pp. 259 – 266, 1998.

[31] G. Lawton, "*Video Streams into the Mainstream*", IEEE Computer Society Press, vol. 33, no. 7, pp. 12-17, 2000.

[32] A. Dantrey, R. Dubey and V. Sharma, "*Exploring Multimedia Transmission Issues over Wireless Channels and their Application in Telemedicine*", B.Tech Project Report, IIT Roorkee, 2008.

[33] T. Bray, J. Paoli, C.M. Sperberg, McQueen, editors, "*ExtensibleMarkup Language (XML)1.0*", W3C Recommendation, 1998.