Title	遠隔教育のための適応型インターネットビデオ送信
Author(s)	Vladimir, Augusto Fabrega Labutina
Citation	
Issue Date	2002-12
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1648
Rights	
Description	Supervisor:Susumu Horiguchi, 情報科学研究科, 修士

Adapting the transmission of visual communication over the Internet for distance learning.

By Vladimir Augusto Fabrega Labutina (010201)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor: Susumu Horiguchi

and approved by
Professor Susumu Horiguchi
Professor Hong Shen
Professor Koichiro Ochimizu

November, 2002

Contents

List of Figures

Acknowledgments

I would like to thank the Japanese Ministry of Education, Culture, Sports, Science and Technology for giving me the opportunity to study my master course in Japan, a great nation.

This master thesis would not have been possible without the constant guidance, inspiration and encouragement of my principal supervisor and thesis advisor, Professor Dr. Susumu Horiguchi. He manages to strike the perfect balance between providing direction and encouraging independence.

Thanks to all my collegues in the laboratory that made this research possible. Thanks to Miss Sarah Gordon for reading and correcting my English.

Finally I would like to thank my family for all their unconditional support.

Abstract

With the recent rapid technological advances and development of the Internet, effective distance learning has become an increasingly plausible alternative to more traditional methods of study. Visual information (such as video (or/and) slides) is proven to be the best way to transmit knowledge. Unfortunately, transmitting video over the Internet is still facing many difficulties, which make visual information transmission inefficient for practical applications. For distance learning, it is even more important to always provide the learner with the requested information under all network conditions. The main goal of this research is to optimize the transmission of visual information by implementing a new protocol that will consider the current status of the network and dynamically provide the requested information. The mission of the protocol is to provide the user with at least three types of reply to the request, depending on the condition of the network. In case of poor network condition, the server will send a pure text-based message describing the contents. If the network condition is fare then a low quality video will be provided. Finally, if the network condition is very good then a high quality video will be transmitted. The proposed protocol is implemented in a server which is connected to the Internet and performance evaluation shows that this dynamic transmission mechanism ensures delivery of the requested information to the end-user under any network conditions. It will also keep the user constantly provided with information from the learning center. Consequently, the distance learning will become increasingly applicable and efficient.

Chapter 1

Introduction

1.1 Background

In distance learning it is important for the users to have access to information when they are ready for it or when they need it regardless of whether the communication between user and server is good or bad. A very important point in this research is to reach the users no matter the overwhelming amounts of information there is on the Internet making the transmission of video over the Internet very difficult. Therefore, the challenge is to present not only visual information but visible information at any time.

The Internet is a network that works only based on the policy of “best effort”. It does not offer any guarantees for the arrival of the packets that are sent from host computer to end-user. This makes it difficult when sending multimedia data because during the transmission of video, we cannot afford to lose too many packets, or the result will be transmission failure.

Until now all the efforts to improve the transmission of video over the Internet have been based on improving the routing algorithms that are used to send packets over the network without considering the type of packets actually been send.

For learning purposes, the most important point is to keep the learner provided with the information requested. This can be achieved by sacrificing the quality of the video. Only providing a high quality video when the network condition is very good. For this reason instead of focusing on how to improve the quality of video, the focus of this research will be on the importance of communicating information that will be relevant in the search for knowledge. For example, when the connection between host and end-user has optimum speed, the user will receive an optimum video quality. But, if the network is congested, the user will receive a video with a lower quality. Finally, if the conditions on the network are congested at any particular moment then the user will receive at minimum a document describing the information requested.

The difference between this research and previous works [1] is that in this research the protocol to be implemented focuses on the need to always provide the user with information even if we have to sacrifice the quality of the video, which is acceptable in applications like distance learning. Instead of ensuring reliable transmission of video packets and improving the quality of the video using different mechanisms (e.g. forward error correction), the focus will be on the end to end adaptation to network congestion, which them can be used to complement error recovery methods.

There are two modes for transmission of stored video over the Internet, namely, the download mode and the streaming mode [12]. In the download mode, a user downloads the entire video file and then plays back the video file. However, full file transfer in the download mode usually suffers long transfer time. On the other hand, in the streaming mode, is not necessary to download the whole video before it starts to be played while parts of the content is being received and decoded. Due to its real-time nature, video streaming typically has bandwidth, delay and loss requirements. However, the current best-effort Internet does not offer any quality of service (QoS) guarantees to streaming video over the Internet. In addition, for multicast, it is difficult to efficiently support multicast video while providing service flexibility to meet a wide range of requirements from the users. Thus, designing mechanisms and protocols for Internet streaming video poses many challenges.

1.2 Objectives

The focus of this research will be on the importance of communicating information that will be relevant for distance learning. For example, when the connection between host and end-user has optimum speed the user will receive an optimum video quality. Nevertheless, if the network is congested, the user will receive a video with a lower quality. Finally, if the conditions on the network are congested at any particular moment then the user will receive at least a document describing the information requested.

To realize the above strategy, the video transmitting protocol changes the quality of the video image according to the condition of the network. Instead of ensuring reliable transmission of video packets and improving the quality of the video using different mechanisms (e.g. forward error correction), the focus will be on the end to end adaptation to network congestion, which then can be used to complement error recovery methods.

In section 2.6 we explain what would be the problem to be solve in this research and how we solve it.

1.3 Organization of this thesis

The rest of this thesis is organized as follows: chapter 2 explains how video transmission is accomplished over the Internet. Chapter 3 describes a new idea on how video should be send over the Internet for distance learning purposes. In Chapter 4 is presented how the server side and the user side are implemented. Chapter 5 some experiments results are presented. Chapter 6 discusses conclusions reach out of this research.

Chapter 2

Video transmission over the Internet

2.1 Outline

In this chapter, section 2.2 explains some concepts in previous works related to transmitting video. Section 2.3 explains distance learning aproaves. Section 2.4 classifies the different methods for transmitting video. Section 2.5 dicuss how a whole video could be downloaded. Section 2.6 talks about streaming a video. Section 2.7 presents a possible problem when streaming video.

2.2 Previous Works

Recent advances in computing technology, compression technology, high bandwidth storage devices, and high-speed networks have made it feasible to provide real-time multimedia services over the Internet. Real-time multimedia, as the name implies, has timing constraints. For example, audio and video data must be played out continuously. If the data does not arrive on time, the playout process will pause, which is annoying to human ears and eyes.

To address these challenges, extensive research has been conducted [1], [12 - 15].

Figure 1 shows an architecture for video streaming. Raw video and audio data are pre-compressed by video compression and audio compression algorithms and then saved in storage devices. Upon the client's request, a streaming server retrieves compressed video/audio data from storage devices and then the application-layer QoS control module adapts the video/audio bit-streams according to the network status and QoS requirements. After the adaptation, the transport protocols packetizes the compressed bit-streams and sends the video/audio packets to the Internet. Packets may be dropped or experience excessive delay inside the Internet due to congestion.

To improve the quality of video/audio transmission, continuous media distribution services (e.g., caching) are deployed in the Internet. For packets that are successfully delivered to the receiver, they first pass through the transport layers and then are processed by the application layer before been decoded at the video/audio decoder. To achieve synchronization between video and audio presentations, media synchronization mechanisms are required. In Figure 1, it can be seen that the six areas are closely related.

There are six key areas with regard to streaming video: video compression, application-layer QoS control, continuous media distribution services, streaming servers, media synchronization mechanisms, and protocols for streaming media. Each of the six areas is a basic building block, with which an architecture for streaming

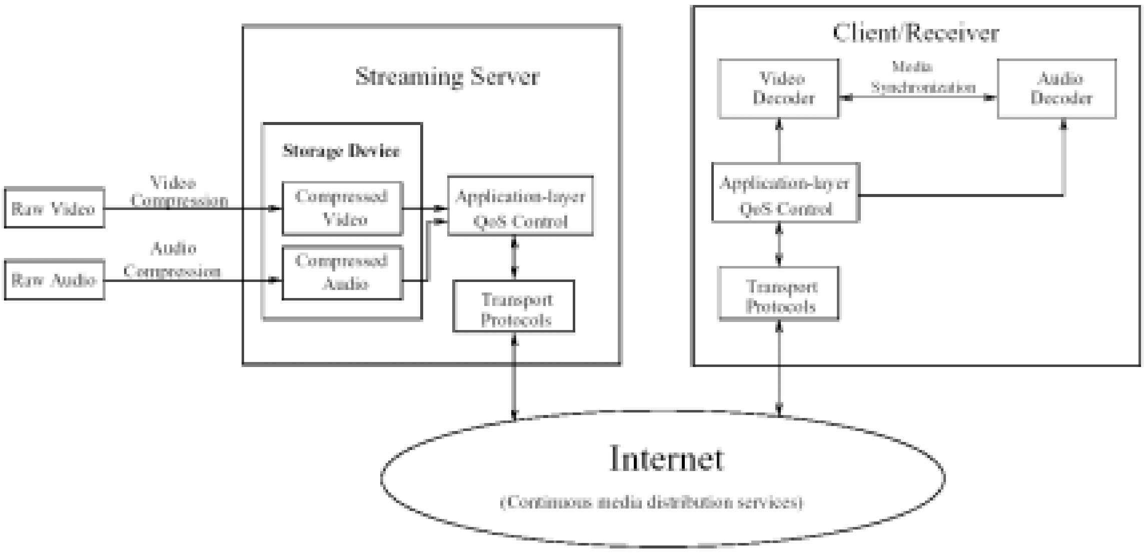

Figure 1 An architecture for video streaming. From reference [12]

video can be built. The relations among the six basic building blocks can be illustrated in Figure 1.

The delivery of streaming media objects presents a formidable strain on server and network capacity due to the long duration and high bandwidth requirements that are characteristic of streaming media workloads. For highly popular streaming media objects, it is especially desirable to utilize truly scalable delivery techniques [16].

The problems posed by heterogeneity are not just theoretical, they impact our daily use of Internet remote-conferencing. For example, each week U.C. Berkeley broadcast a seminar over their campus network and onto the Internet. As depicted in Figure 2 , a video application is run on a “seminar host” that sources a single-rate signal at 128 kb/s, the nominal rate for video over the Internet Multicast Backbone, or Mbone [17]. However, a number of users on the local campus network have high bandwidth connectivity and would prefer to receive higher-rate, higher-quality video. At the other bandwidth extreme, many users have ISDN access and would like to participate from home, but a 128kb/s video stream overwhelms an ISDN line.

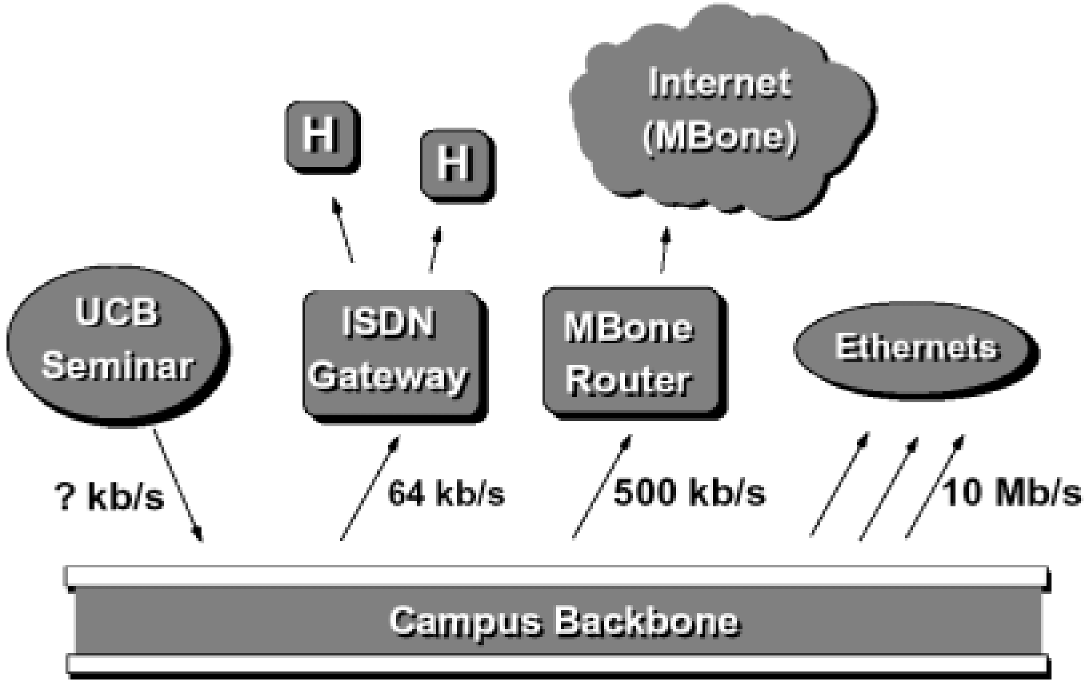

Figure 2 Network heterogeneity [17].

2.3 Distance learning

Distance learning is especially suited for busy people who wish to increase their knowledge and skills. With distance learning, you do not have to give up your job, leave home or lose income. You learn as you earn. The school comes to you. You receive individual attention, and often you work at your own pace. Emphasis is on learning what you need to know. Instructional materials are up-to-date, clearly written and easy to understand [20].

As new technologies developed, distance instruction was delivered through such media as audiotape, videotape, radio and television broadcasting, and satellite transmission. Microcomputers, the Internet, and the World Wide Web are shaping the current generation of distance learning, and virtual reality, artificial intelligence, and knowledge systems may be next. Some define distance education as the use of print or electronic communications media to deliver instruction when teachers and learners are separated in place and/or time [22].

Advantages of delivering distance learning on the Internet include the following [22- 24]. (1) time and place flexibility; (2) potential to reach a global audience; (3) no concern about compatibility of computer equipment and operating systems; (4) quick development time, compared to videos and CD-ROMs; (5) easy updating of content, as well as archival capabilities; and (6) usually lower development and operating costs, compared to satellite broadcasting, for example. Carefully designed Internet courses can enhance interactivity between instructors and learners and among learners, which is a serious limitation of some DL formats. Equity is often mentioned as a benefit of online learning; the relative anonymity of computer communication has the potential to give voice to those reluctant to speak in face-to-face situations and to allow learner contributions to be judged on their own merit, unaffected by "any obvious visual cultural markers" [23]. The medium also supports self-directed learning--computer conferencing requires learner motivation, self-discipline, and responsibility [21].

[25] Studied the effectiveness of a mix of audio plus video in corporate training. When he introduced real-time interactivity, the retention rate of the trainees was raised from about 20 percent (using ordinary classroom methods) to about 75 percent.

2.4 Classification of the methods to send video over the Internet

Three developments made video on the web possible: advanced compression techniques, which made file size dramatically smaller; streaming techniques, which made it possible to begin playing files long before the entire file had completed its download; and higher bandwidth, which established a larger "pipe" between the individual computer and the Internet and allows more information to be transferred more quickly.

There are two modes for transmission of stored video over the Internet, namely, the download mode and the streaming mode [12]. In the download mode, a user downloads an entire video file and then plays back the video file. However, full file transfer in the download mode usually suffers long transfer times. On the other hand, in the streaming mode, it is not necessary to download the whole video before it starts playing and while parts of the content are being received and decoded. Due to its real-time nature, video streaming typically has bandwidth, delay and loss requirements. However, the current best-effort Internet does not offer any quality of service (QoS) guarantees to streaming video over the Internet. In addition, for multicast, it is difficult to efficiently support multicast video while providing service flexibility to meet a wide range of requirements from the users. Thus, designing mechanisms and protocols for Internet streaming video poses many challenges.

2.5 Transmitting a whole file

To transmit a whole file a user downloads the entire video file and then plays back the video file. However, full file transfer in the download mode usually suffers long transfer times because video files are usually very large.

Compression Techniques: The first factor that initially made video on the web prohibitively slow was file size. A standard Windows video file (in AVI format) of a one-hour ethics lecture might be almost 700 MB in size--so large that it would take hours to download even on fast connections. Video over the Web only became practical when it became possible to reduce the size of these files dramatically. RealVideo led the way in the development of the compression techniques that led to startlingly smaller files. A one-hour video, previously 700 MB in a native video format, could be reduced to 5% of its original size--approximately 3.5 MB. This is still quite a large file, but it is at least manageable.

2.6 Streaming the file

Streaming allows digital video and audio to be sent to a user's computer so that the user gets an uninterrupted flow of data. Video streaming involves the short-term storage of media files on the user's machine as compared to the downloading of large multimedia files. When streaming a video the end user does not have to wait to download a whole file to start watching its contents.

Streaming Techniques: The second factor that made video over the Web possible was the development of streaming techniques. Initially, when video was put on the Web, it was very slow in part because the entire video file had to be downloaded before any of it could be played. The advent of streaming media changed all that, because it made it possible to begin playing a file before the whole file was downloaded. Basically, streaming media starts playing the first chunk of a file (called a packet) while at the same time it is reaching out and grabbing the next chunk to be played. The result is that streaming video begins playing almost immediately--within the 10-20 seconds it takes to download the initial packets and to get slightly ahead in downloading additional packets. This process of getting the next packets lined up ahead of time is called “buffering”.

Higher Bandwidth Access: These developments in video technology were augmented by the development of faster access to the Internet. Some of this has been due to increases in modem speed. At one time, 14.4 Kbps was considered fast; then the standard rose to 28.8Kbps, and now 56Kbps is the standard speed for modems. (The amount actually transmitted, the actual throughput, is typically less than the maximum, and a 56K modem may only deliver 40K of actual throughput.) The most dramatic increases, however, have resulted from getting away from traditional modems. Digital Subscriber Lines (DSL) and cable modems (running over the local cable television fiber optic lines) have dramatically increased the bandwidth, (e.g. the amount of information that can be transmitted in any given second). At colleges and universities as well as at corporations, computers typically access the Internet through a local area network (LAN) that is directly hooked up to the Internet over a very high-speed connection (a T1 or T3 line). College dorm rooms now are usually wired to allow such fast connections to the Internet, and increasingly students are reluctant to move off campus in their junior and senior years because it means giving up their Internet connection.

2.7 Explanation of the problem to be solved

As we have seen in the previous section downloading a video file could take hours even with a fast connection. Initially, when video was put on the Web, it was very slow in part because the entire video file had to be downloaded before any of it could be played.. Therefore, in this research it is considered that streaming a video file is a better solution, because, it made it possible to begin playing a file before the whole file was downloaded.

2.7.1 Why streaming could be a problem?

Although streaming is the best way to transmit video over the Internet it could become a problem when the bandwidth in at particular moment is congested. At that time, the video would stall and it would stop or became very slow as shown in Figure 3.

Figure 3 shows how streaming could be a problem. Because depending on the bandwidth situation the server will drop more frame when the bandwidth is congested therefore the user would have a video that will present frame that are not in succession but very much apart, the video will stall and will not be satisfactory for the end user. For example as you see in the server side the video has all its frames from 1, 2, 3 until N, but in the user side a lot of frames has been drop, like 2, 3, 4, etc. And the more congested the network is the more frames are drop.

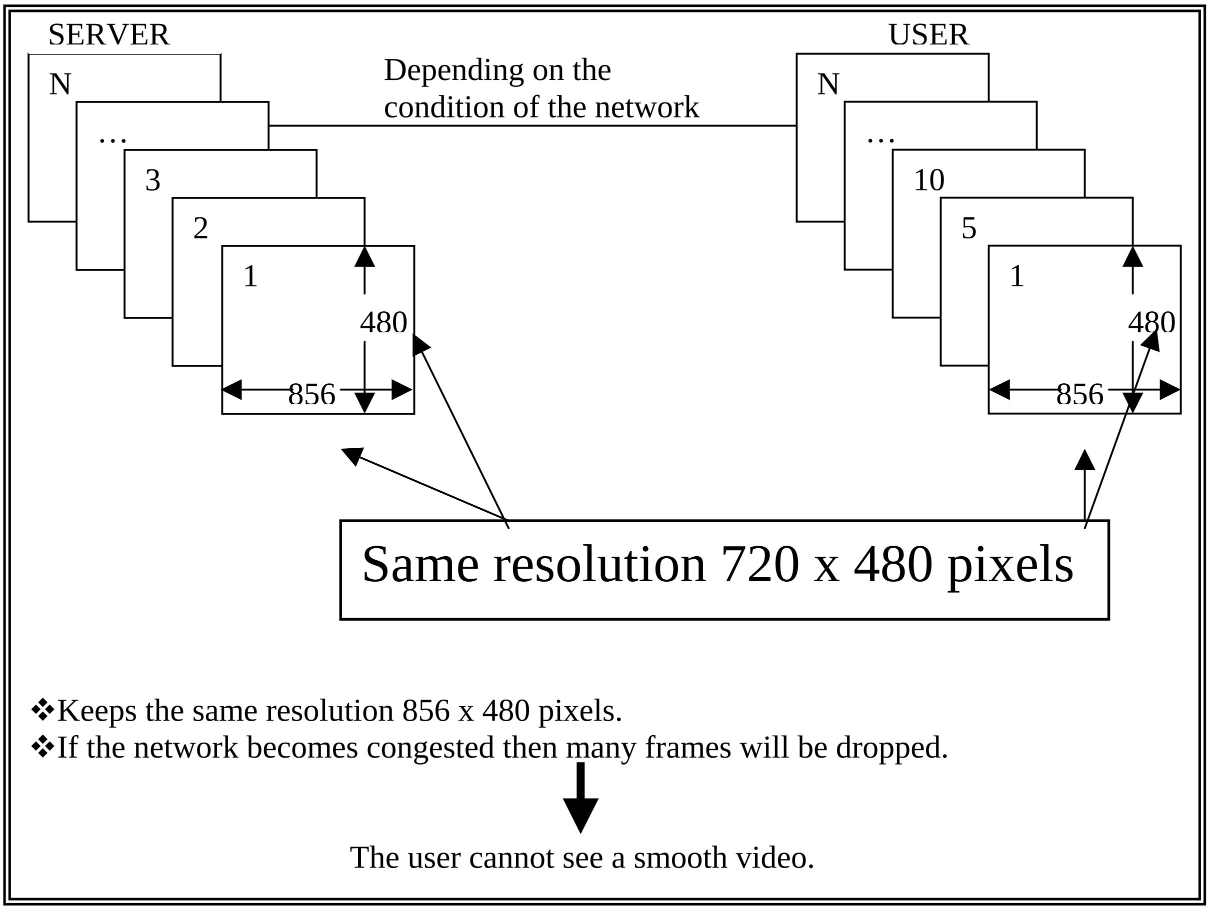

Figure 3 Streaming video with the same resolution cause some problems

2.7.2 A new approach to streaming video

On the other hand to solve this problem we can use different types of resolution of the same video. Therefore, if the bandwidth is congested you receive a small resolution video and if it is bigger, you receive a bigger resolution video, as shown in Figure 4. Thus obtaining the advantage of using less bandwidth and at the same time dropping some frame is necessary as previous approaches do.
As you can see in the example the way this research solve the problem of smoothness when the condition on the network is congested is by storing two videos with the same presentation on the server size. That way when transmitting the over the network the amount of frames that will be drop will be much less as in previous approaches, in this example just frame 2 is drop.

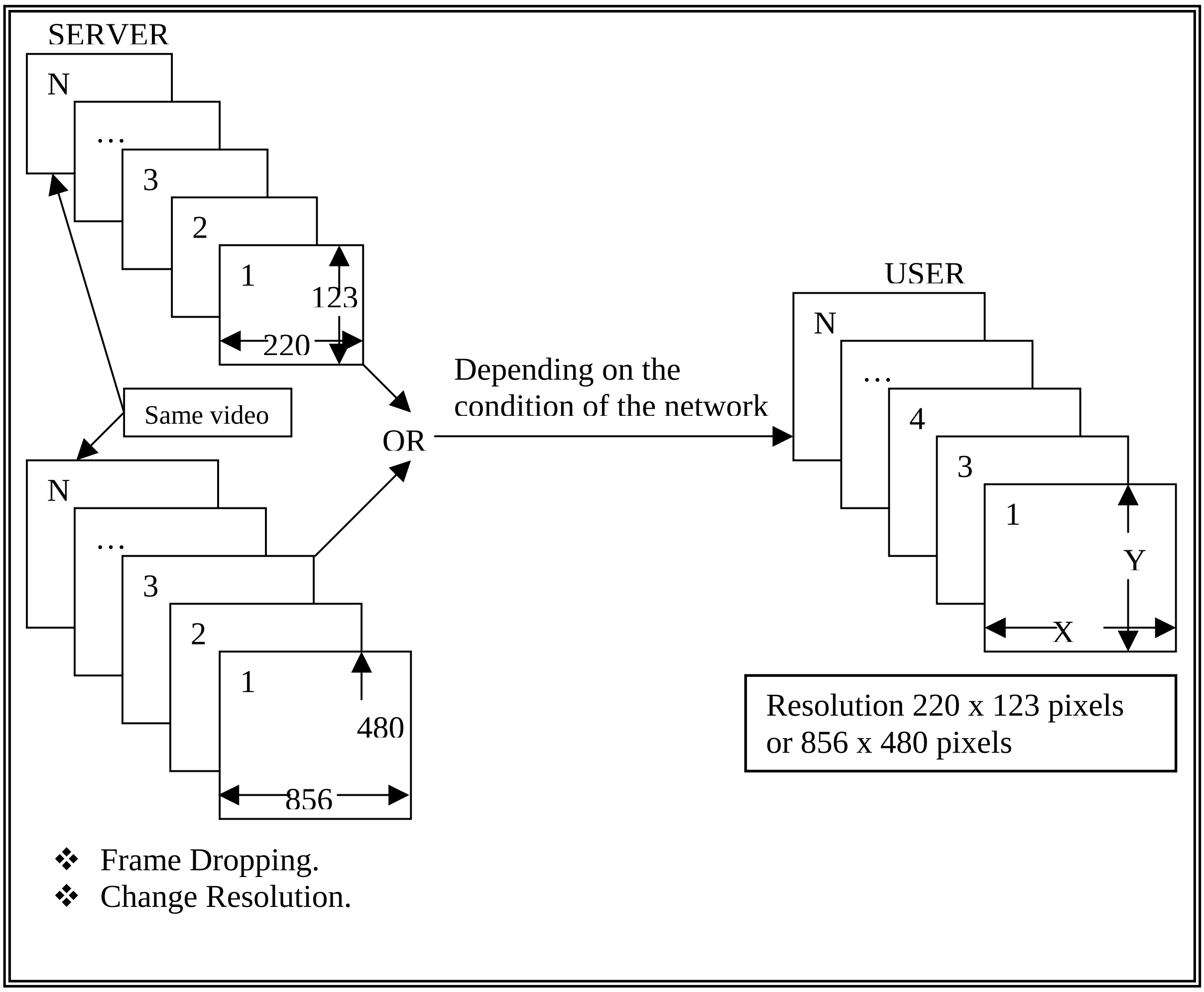

Figure 4 A new approach to streaming video.

Chapter 3

Adapting the transmission of video over the Internet for distance learning purposes

3.1 Outline

This Chapter is divided into two parts, the first part, section 3.2, explains previous work using different protocols and the second part, section 3.3 explains the new approach to sending video for distance learning over the Internet. The second part is also divided into two parts. First, section 3.3, is an outline of the protocol providing a general explanation of how the protocol works, and secondly, section 3.7, is a more detailed explanation. This includes a section regarding sending a ping (section 3.4). Explanation of the structure of that ping (section 3.5). The uses for the information retrieved (section 3.6). The resulting information sent depending on the information that the ping retrieves (section 3.7). Section 3.8 discusses the policy implemented in this protocol. Explanation on how the video data is collected and compress (section 3.9). And finally, section 3.10 presentation of the source code.

3.2 Previous protocols

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 (RTSP).

Some protocols have been designed and standardized for communication between clients and streaming servers. The protocols related to Internet streaming video can be classified into the following three categories: Network-layer protocol, Transport protocol, Session control protocol.

3.2.1 Network-Layer protocol

Network-Layer protocol provides basic network service support, for example, network addressing. The Internet protocol (IP) serves as the network-layer protocol for streaming video over the Internet for distance learning.

3.2.2 Transport protocol

Transport protocol provides end-to-end network transport functions for streaming applications. Transport protocols include UDP, TCP, real-time transport protocol (RTP), and real-time control protocol (RTCP). UDP and TCP are lower-layer transport protocols while RTP and RTCP are upper-layer transport protocols, which are implemented on top of the UDP/TCP protocols [13]. See Figure 5.

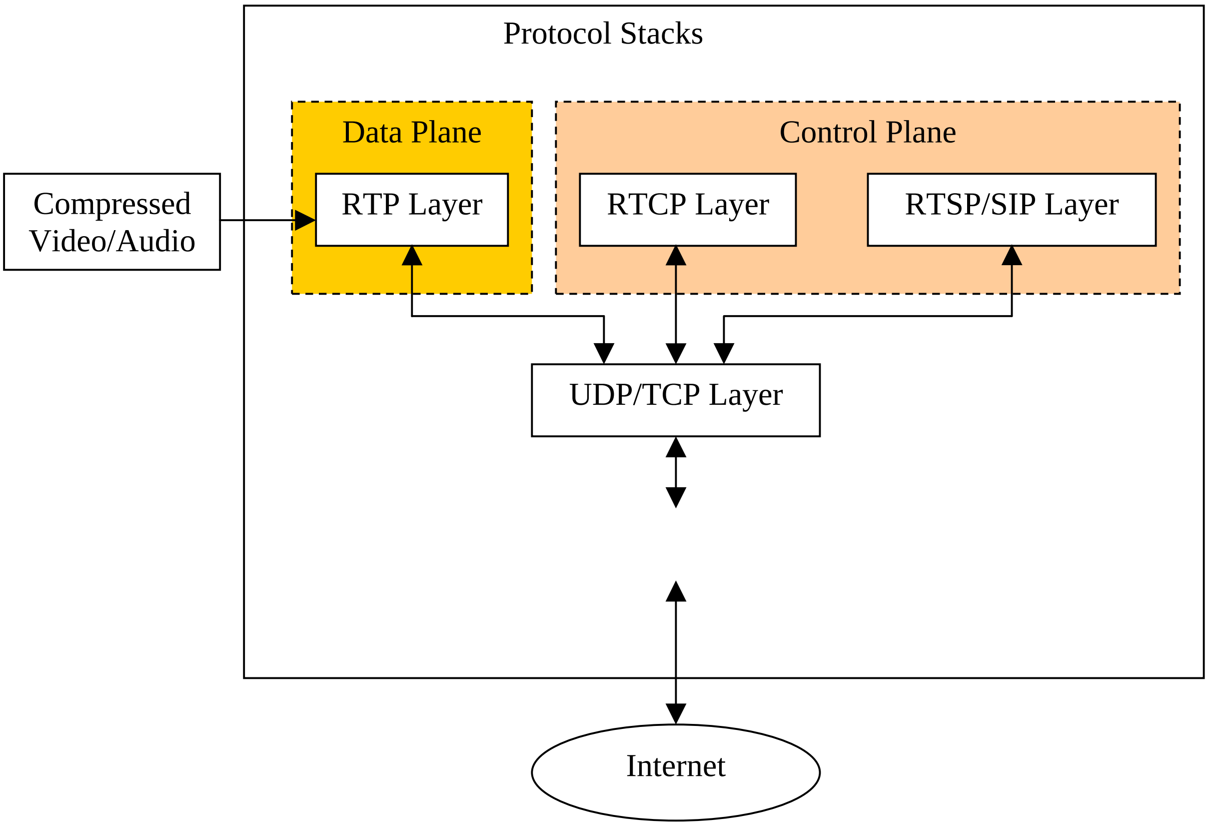

Figure 5 Protocol stacks for media streaming [12]

3.2.3 Session control protocol

Session control protocol defines the messages and procedures to control the delivery of the multimedia data during and established session. Among this protocols are the real-time streaming protocol (RTSP) [14] and the session initiation protocol (SIP) [15]

The relationship among the three types of protocols is illustrated in Figure 5. At the sending side, the compressed video/audio data is retrieved and packetized at the RTP layer. The RTP-packetized streams provide timing and synchronization information, as well as sequence numbers. The RTP-packetized streams are then passed to the UDP/TCP layer and the IP layer. The resulting IP packets are transported over the Internet. At the receiver side, the media streams are processed in the opposite direction before they are presented. In the control plane, RTCP packets and RTSP packets are multiplexed at the UDP/TCP layer and move to the IP layer for transmission over the Internet.

3.3 Outline of the new protocol

The new system being implemented does not fall into any of the three categories mentioned before because it is actually an end-to-end solution that runs using all of the mentioned protocols to transmit video over the Internet for distance learning.

As explained in the introduction of this research the way this protocol works is as follows:

First, a ping packet is sent from the server to the end user and back to determine the quality of communication between the two. See Figure 6.

3.4 Initial step (Sending a ping)

A ping is used to determine whether a specific IP address is accessible. It works by sending a packet to the specified address and waiting for a reply. Ping is used primarily to troubleshoot Internet connections.

Pings are one of the most useful network debugging tools available. They take their name from submarine sonar searches. A short burst of sound is sent and then the echo is listened for. The sound produced by the echo is a “ping”.

In an IP network, a ping sends a short data burst: a single packet, and waits for a single packet in reply. This test is the most basic function of an IP network (delivery of a single packet).

Pings are implemented using the required ICMP Echo function, documented in [18] that all hosts should implement. Of course, administrators can disable ping messages (this is rarely a good idea, unless security considerations dictate that the host should be unreachable anyway), and some implementations have even been known not to implement all required functions. However, pings are usually a better bet than almost any other network software.

Many versions of ping are available.

3.4.1 What a ping can show

A ping places a unique sequence number on each packet it transmits, and then reports the sequence numbers it receives back. Thus, it can be determined if packets have been dropped, duplicated, or reordered.

- The ping checksums each packet it exchanges. It can detect some forms of damaged packets.

- Pings place a timestamp in each packet, which is echoed back and can easily be used to compute how long each packet exchange took this is known as The Round Trip Time (RTT).

- Additionally a ping reports other ICMP messages that might otherwise get buried in the system software. It reports, for example, if a router is declaring the target host unreachable.

3.5 Structure of the ping packet to be sent

ping -s –U users_ip [data_size] [count]

3.5.1Explanation of the ping packet structure

In this research the structure used for the ping packet is as follows:

- -s : sends one data gram per second and collect statistics.

- -U: sends UDP packets instead of ICMP packets. In order not to use the TCP/IP channel that would be used to transmit the rest of the information.

- users_ip: this is the IP address of the user: the person that makes the request.

- data_size: to simulate different sizes of data when sending the ping.

- count: to send the ping more than once in order to get an average of the values that would be used to decides the quality of communication.

3.6. Information retrieved from using a ping

The values that the ping provides are "packet loss" and "average". "Packet loss" is how many packets were lost when sending the ping and "Average" is an average of the time in milliseconds that the packets take to go from the server to the user and back. These information are used by the protocol to decide which video to send as shown in Figure 6.

Depending on the connection at that particular moment an appropriate video or quantity of information is sent. This decision is made based on the quantity of packet loss the ping registered. And the values are shown in Table 1 [2], [3].

Packet Loss (%)	Rank
0 – 1	Good
1 – 2.5	Acceptable
2.5 – 5	Poor
5 – 12	Very Poor
> 12	Unusable

Table 1. The condition of the network depending on the packet loss registered by the ping [2]

The response-Time in milliseconds is also considered, in order to decide which video will be sent. See Table 2 [2], [3].

Response Time (ms)	Rank
0 – 62.5	Good
62.5 – 150	Acceptable
150 – 250	Poor
250 – 500	Very Poor
> 500	Unusable

Table 2. **T**he condition of the network depending on the Response-Time (in milliseconds) registered by the ping [2]

For example, when the connection between host and end-user has optimum speed the user receives an optimum video quality. But, if the network is congested then the user receives a video with a lower quality. And finally if the condition of the network is too congested at any particular moment then the user will receive at minimum a document describing the information requested. See Figure 6.

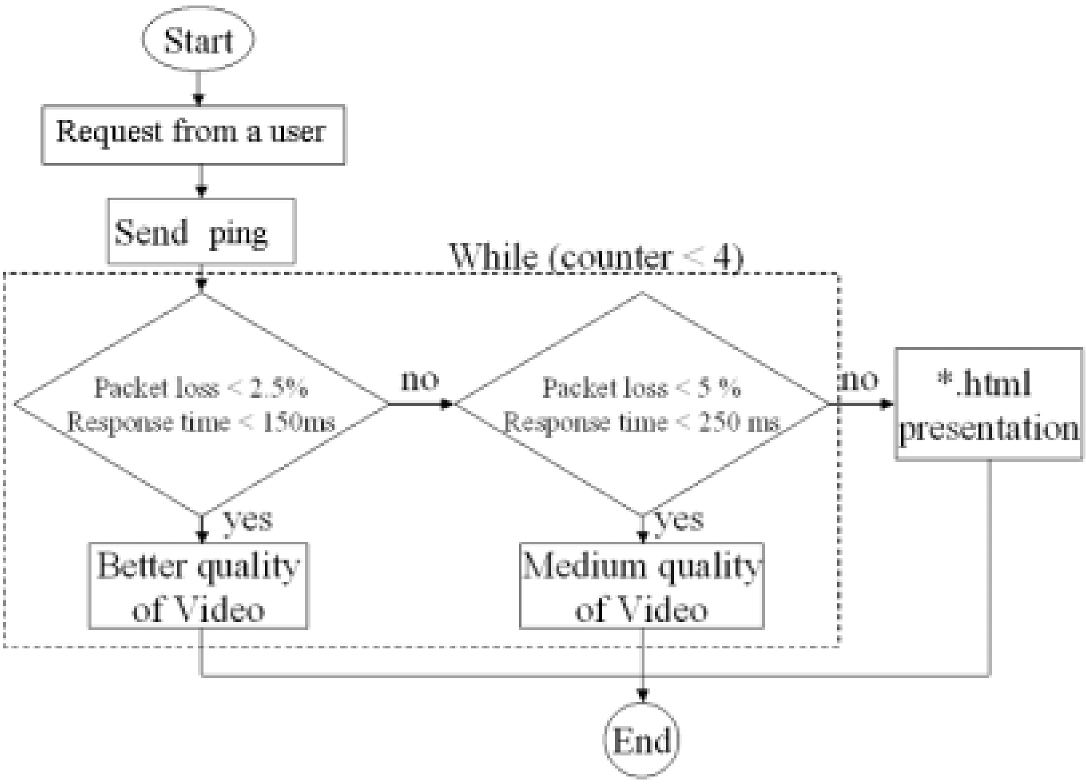

Figure 6 Flowchart showing the implemented protocol.

In Figure 6, First the user side request to see a video, at that moment the server will send a ping once it has optain this information the server will deside depending on “packet loos” and “response time” what type of video will send, one with better quality or one with medium quality. Or if the network condition in not good enough then in will send a text information of the presentation. This will be done every 5 minutes, so in a 20 minutes presentation this procedure will be done 4 times.

3.7 Explanation of the proposed protocol

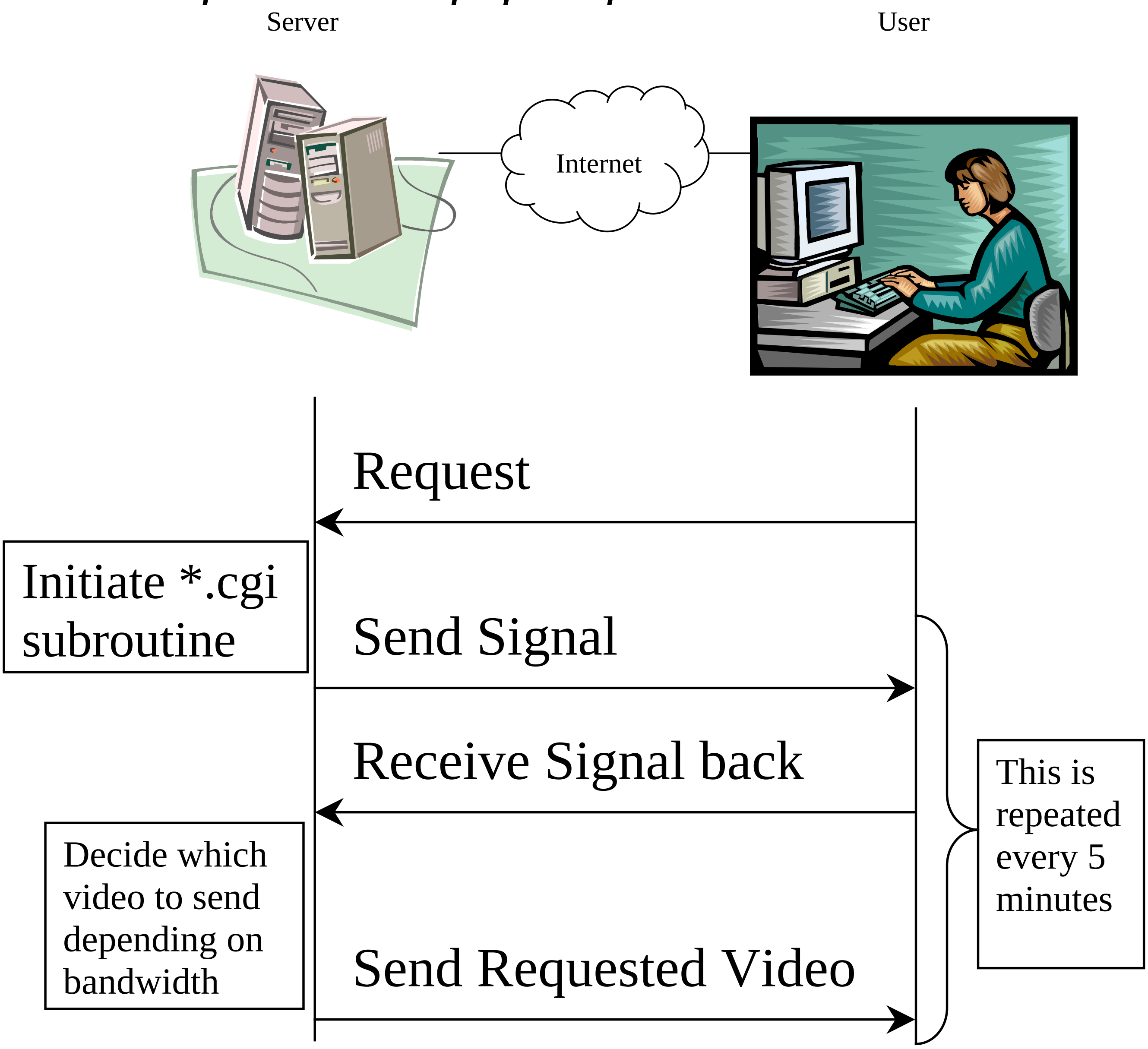

Figure 7 How the proposed method works.

To explain the proposed protocol Figure 7 should be studied. First, the user makes a “request” to watch a video. After the server receives this “request” it will start a subroutine. This subroutine will first send a signal (a ping) to the user and this signal will travel back to the server with information on the state of the network at that specific moment. With the information retrieved, packet loss and response time (see Tables 1 and 2), the server will decide which information to send. The information could be a presentation in a written format of the presentation if the network condition is bad, or it could be a medium quality resolution video (220 x123) if the conditions are moderate and finally it could be a high quality resolution video (856 x 480) if the condition of the network is good. See Figure 6.

3.8 Policy to decide which video to send

As already stated the Internet does not offer QoS, thus the condition of the network could change at any time and thus does not offer any assurance that the same bandwidth between the server and the user will remain constant. Therefore, it is preferable that the videos with medium quality and optimum quality are divided into 4 parts. Then each time one of the video parts finishes then the protocol can re-assess the condition of the network so that the server can decide to change the quality of the video if the network conditions have changed. See Figure 8.

	First Part	Second Part	Third Part	Fourth Part
Better quality	5 min	5 min	5 min	5 min
Medium Quality	5 min	5 min	5 min	5 min
*.PPT				

Figure 8 Policy determining which information to send

3.9 Collection and compression of the information

As part of this research, a server is installed to transmit video over the Internet for distance learning. The installation of the real player server is explained in chapter 4. The experiment material (data) that is transmitted from that server is the seminars that are hosted in our lab.

Therefore, first the presentation of the seminar in our lab is video-taped, and that information is stored in the server, and as part of the research which compression is the best compression in terms of file size and streaming capabilities will be considered.

After storing this information in the form of video in the server with the best possible compression a new protocol is implemented. This new protocol decides for the user, which file to retrieve from the server depending on the network capabilities at that moment. For example, when the connection between host and end-user has optimum speed the user receives an optimum video quality resolution (example 856 x 480). But, if the network is congested then the user receive a video with a lower quality resolution (example 220 x 123). And finally if the conditions on the network are too congested at any particular moment then the user will receive at minimum a document describing the information requested. The parameters used to determine the condition of the network will be explain in a later chapter.

3.9.1 Storing the information in the form of videos and slides onto a server

The method used in this research to store the videos and slides onto the server is as follows:

- Using either canopus (video card) program DV-Storm-RT or Adobe Premier 6.0 to store the videos on the server in *.avi format.

- After storing these videos in *.avi format then they are compressed them using Adobe Premier 6.0 and Cleaner EZ 5.0 to transform these files into MPEG format. Figure 9 shows the connection between camera and computer.

- Finally, to store the presentation on slide on to the server it is just necessary to copy the file *.ppt file directly onto the server.

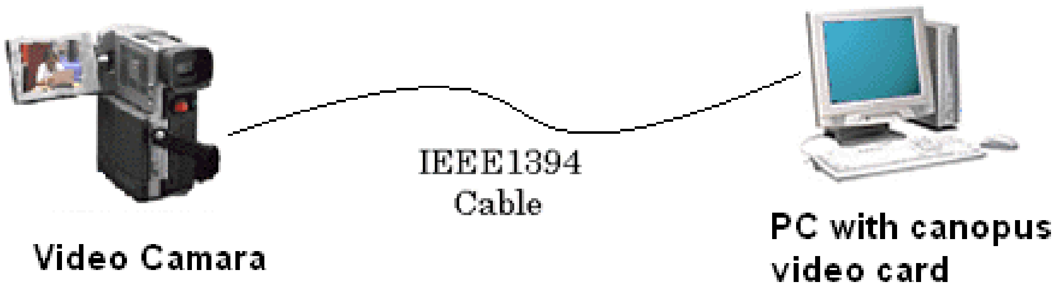

Figure 9 Shows a video camera connected with an IEEE1394 connection to a computer that has a canopus video card.

3.9.2 Compression of the video files

An objective in this research is to find out which video compression is better for streaming and storing the video depending on the size of the file.

A raw *.avi files take too much disk space, e.g. a file which is approximately 19 minutes of video will take 4.1GB of space. Meanwhile the same file of 19 minutes will take 198 MB of space if this file is in MPEG format. This is still quite big and a better compression rate will need to be found. On the other hand the same file takes 15 MB in *.rm format, which is quite acceptable. See Figure 10.

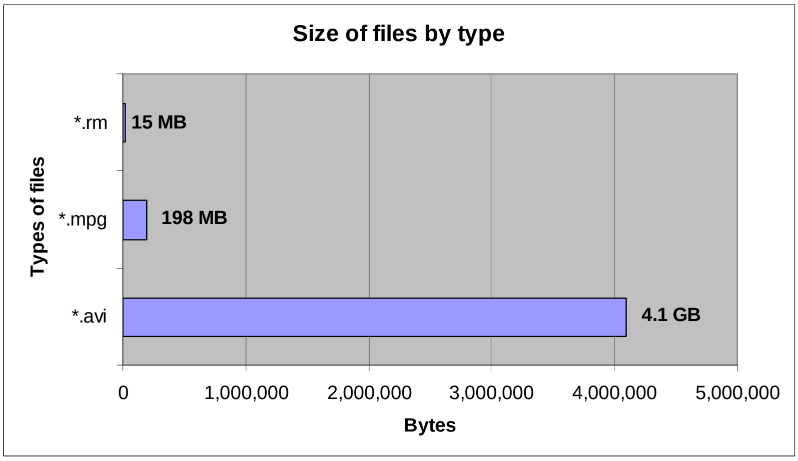

Figure 10 Shows the size in bytes of a 19 minutes file depending on the type of compression.

3.10 Source Code

This Source code is the implementation of the protocol proposed in this reseach.

```
#!/usr/local/bin/perl
$ping = "/usr/sbin/ping";            # Location of the ping programan

&Main();

sub Ping {
$domain = "$ENV{'REMOTE_ADDR'}";      # Client IP address

$pingdata = `$ping -s -U \"$domain\" \"23\" \"2\"`; # Ping information

$index = index($pingdata, received) + 10;   #find where is the string %
$packetloss = substr($pingdata, $index, 1); #substract the value of packet loss

$index = index($pingdata, "max = ")+6;    #find where is the string
$substr1 = substr($pingdata, $index);
@average = split(/\//,$substr1);

$otherpacket = $packetloss;
$otheraverage = $average[1];

}

sub Main {

&Ping();

if ( $packetloss > 50 && $average > 250){
print "Content-type: text/html\n\n";
print <<EndHTML
   <html><head><title>Show Written Presentation</title></head>
   <body>
   <h2>Hello!</h2>
Welcome, visitor from: $ENV{'REMOTE_HOST'}
   <p>
Your IP Address is: $domain
   <p>
$pingdata
   <p>
packetloss = $otherpacket
   <p>
```

```
average = $otheraverage
   <p>
Server Port: $ENV{'SERVER_PORT'}
   <p>
   At the present moment there is to much traffic on the network to receive
video.
   <a href="http://www.jaist.ac.jp/~fabrega/movie/eisuke8-3-2002.htm">
   Click here if you would like to open a *.ppt version of this file</a>
   </body></html>
EndHTML
}
else {
if ( $packetloss <  26 && $average < 151) {
     $video = "http://www.jaist.ac.jp/~fabrega/cgi-bin/mediumfirstpart.ram";}
elsif ( $packetloss > 25 && $packetloss < 50 && $average > 150 && $average <
250){
     $video = "http://www.jaist.ac.jp/~fabrega/cgi-bin/qualityfirstpart.ram";}

print "Content-type: text/html\n\n";
print <<EndHTML
<html>
<SCRIPT LANGUAGE="JavaScript">
<!--
setTimeout("location.href='$video'",0000);
-->
</SCRIPT>
</html>
EndHTML
    }

}
```

Chapter 4

Implementation of the Adaptive Video Transmission System

4.1 Outline

In this chapter section 4.2 explains the server side. Section 4.3 presents the user side. Section 4.4 discusses how the server side and the user side communicate. And finally, section 4.5 explains other tools used in this research.

4.2 Server side

Although it is not the intention of this research to explain in depth how to install a Real Server in this section, we describe briefly how to accomplish this and how to use some of the other tools used in this research to obtain the results of the experiments.

Just as a Web server delivers pages to Web browsers over the Internet, our video server serves media clips to clients. It enables users to stream the media clips rather than download them. By streaming the content, the user can begin to watch the clip almost immediately and doesn't have to wait for the entire file to be downloaded.

4.2.1 What Is Real Server?

Real Server is server software that streams both live and prerecorded media over a network. The streamed data can originate either on the Internet or within an intranet. The client receives the media in real time, and without having to wait for clips to be downloaded.

Real Server streams media to clients over networks and the Internet. It is usually employed in conjunction with a Web server.

4.2.2 Real Server components

- **Executable file:** Real Server’s main component, called rmserver.exe for Windows-based platforms and rmserver for UNIX-based platforms.

- **Plug-ins:** Files that provides the functionality of Real Server's individual features. Because of this open architecture, third parties can create custom features, enabling you to extension of the capabilities of Real Server.

- **Configuration file:** A text file in XML format that stores all of your Real Server's customized information. The configuration file is named rmserver.cfg.

- **License file:** One or more files that control the features enabled in Real Server.

- **Real System Administrator:** A Web-based console for customizing and monitoring Real Server.

- **Tools: A**dditional softwares such as the Java Monitor, which enables the view of the number of clips that are being served at any given time, and G2SLTA, which broadcasts prerecorded clips as if they were live events.

- **Other files:** This depends on the particular Real Server package that have been installed. The installation may have other files that perform additional functions, such as commerce or ISP hosting.

4.2.3 What Is Real System?

Real Server is a member of the Real System family of software tools. Three components make up Real System:

- **Production tools-**such as Real Producer Plus, to create media clips (such as audio, video, or animation). These are also called *encoders*.

- **To-**to stream media files.

- **Client software-**such as Real Player, to play the streamed clips.

Figure 11 illustrates how these Real System components work together.

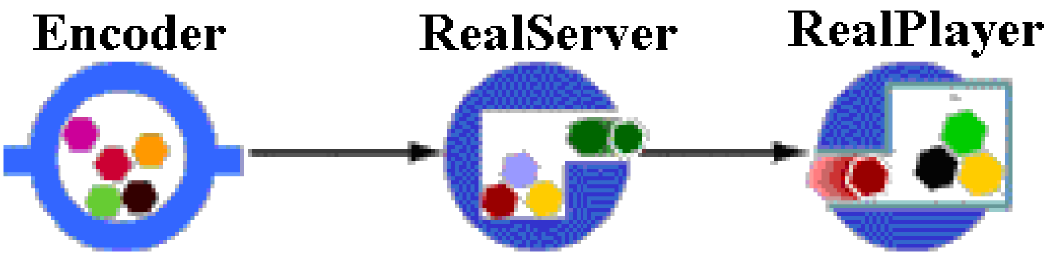

Figure 11 Real System Components

In this research Adobe Premier 6 was used as Encoder, but Real Server 8.0 and different versions of Real player were used.

4.2.4 Channels and Protocols

Real Server uses two connections, known as *channels*, to communicate with clients: one for communication with the client, and one for actual data. The communication channel is known as the *control channel*, as it is over this line that Real Server requests and receives passwords and clients send instructions such as fast-forward, pause, and stop. Media clips themselves, on the other hand, are actually streamed over a separate *data channel*. Every link to content begins with a protocol identifier, such as rtsp, pnm, or http.

Real Server uses two main protocols to communicate with clients: Real Time Streaming Protocol (RTSP) and Progressive Networks Audio (PNA).

Occasionally, Real Server will use HTTP for metafiles that point to Real Server content, and for HTML pages that it serves (such as the Web-based Real System Administrator). It may also be used to deliver clips to clients that are located behind firewalls. Within these channels, Real Server uses two other protocols for sending instructions and data:

Transport Control Protocol (TCP): For sending commands from the client (such as "start" and "pause") and sending commands from Real Server to clients for specific information (such as the clips' titles).

User Datagram Protocol (UDP): For sending actual streamed content.

4.3 User side

A client such as Real Player plays the streamed media. Please see Figure 12.

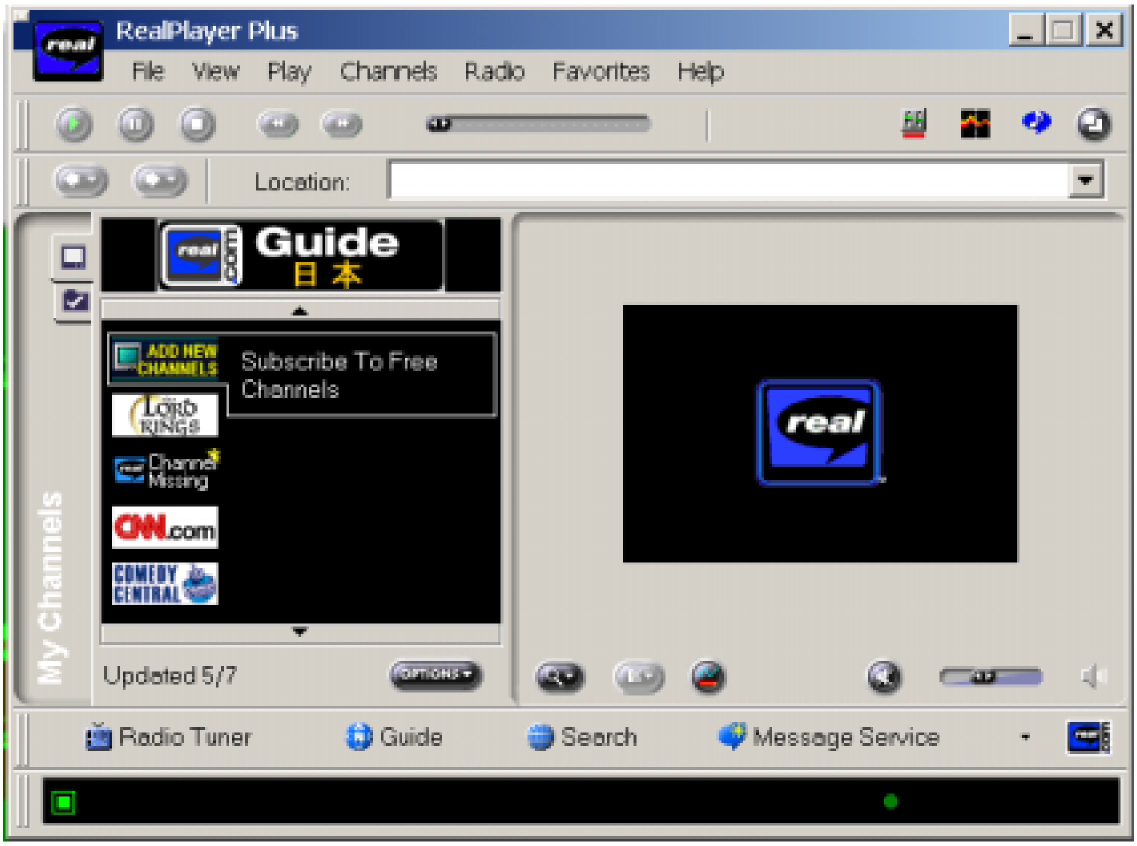

Figure 12 Real Player Plus

4.4 Communication between server and users

When the user clicks a link that points to a streaming media presentation, Real Player opens a two-way connection with Real Server. This connection uses TCP to send information back and forth between Real Player and Real Server as shown in Figure 13.

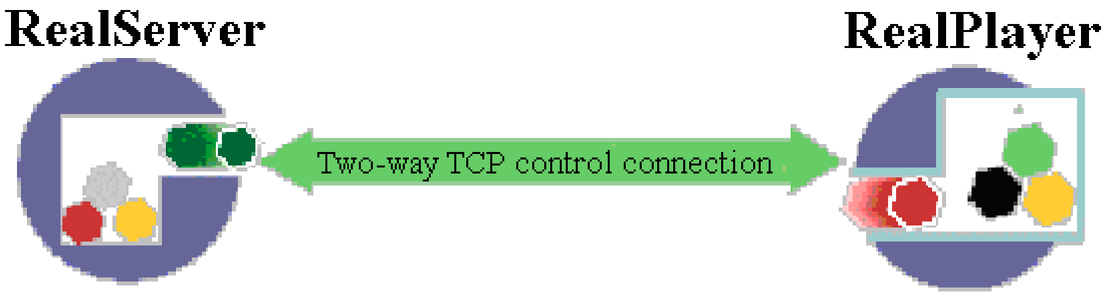

Figure 13 Initial TCP Control Connection

As Real Server approves the request, it sends the requested clip to a Real Player along a one-way UDP channel, as shown in Figure 14 .

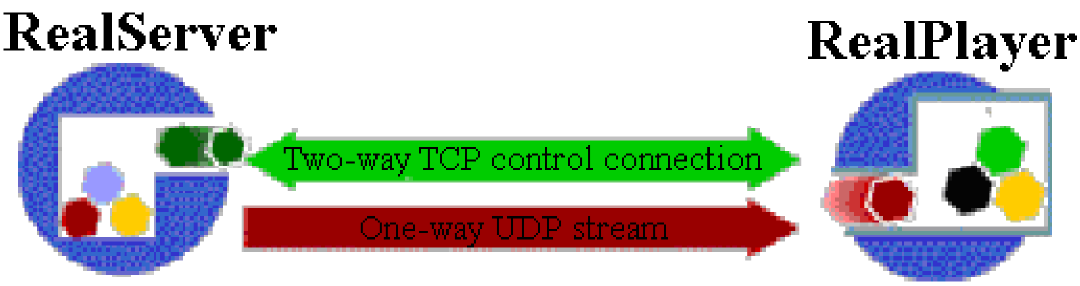

Figure 14 UDP Data Connection

As it receives the streamed clip, Real Player plays it at high fidelity.

4.5 Other tools used in this research

4.5.1 Bandwidth Simulator

In order to simulate different bandwidth we used bandwidth simulator, this tool is an extra tool that can be attach to Real player.

4.5.1.1 How Bandwidth Simulator Works

The Bandwidth Simulator can be started from Real Player. This product allows the user to simulate video playback for a range of Bandwidth conditions. The Bandwidth Simulator works with Real Player to give the user an accurate look at what the presentation will look like using different bandwidths. Only presentations streamed from a Real Server or HTTP server can be used in the Bandwidth Simulator.

4.5.1.2How to start the Bandwidth Simulator

1. Open Real Player

2. In Real Player 8 go to View menu and then click on Bandwidth Simulator, or in Real Player One go to Tools and click on Bandwidth Simulator, these will start the tool, please see Figure 15.

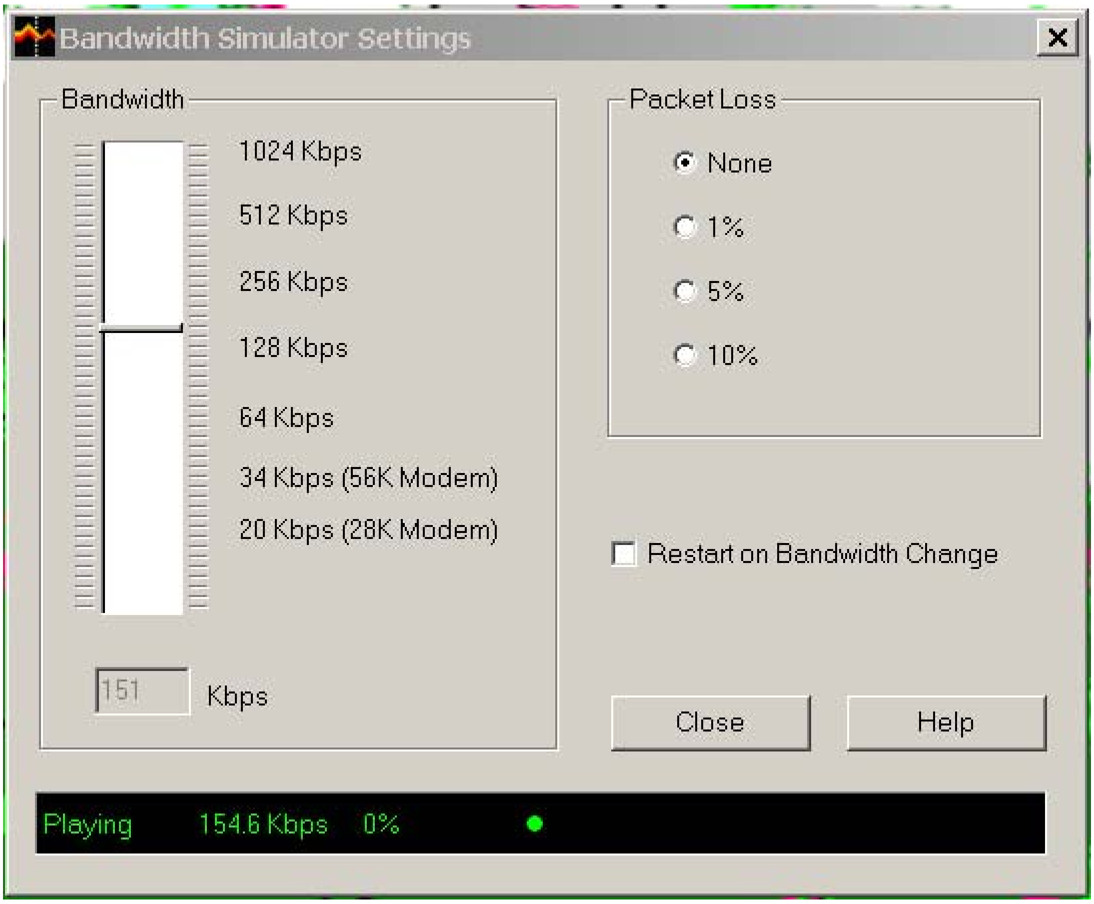

Figure 15 Bandwidth Simulator

4.5.1.3 Using the Bandwidth Simulator

The simulator works only when playing a Real Media presentation from a server. Follow the procedure below to run the simulator properly:

1. Slide the Bandwidth slider to the bit rate at which you want to test the presentation. In this research 15, 20,34,64, 99 and 1024 Kbps were used.

2. Select the amount of packet loss you want to simulate.

   These settings force the simulator to drop data packets during playback, simulating a poor connection. The top range is 10%, a severely bad connection. In the experiments this would not be simulated.
3. Keep the Bandwidth Simulator open, open and begin playing the video using Real Player.

4. Using Statistics monitor the playback (condition or quality) in Real Player.

4.5.2 Statistics

In order to monitor the quality of the video playback statistic was used. This is a tool that comes with Real player.

4.5.2.1 How Statistics Works

The Statistic tool is started from Real Player. This tool allows the user to monitor the video playback. It gives the user accurate information on what Bandwidth is been used, bit rate, frame per seconds, along some other information.

4.4.2.2 How to start the Statistics

1. Open Real Player

2. In Real Player go to View menu and then click on “Statistics”, these will start the tool, please see Figure 16.

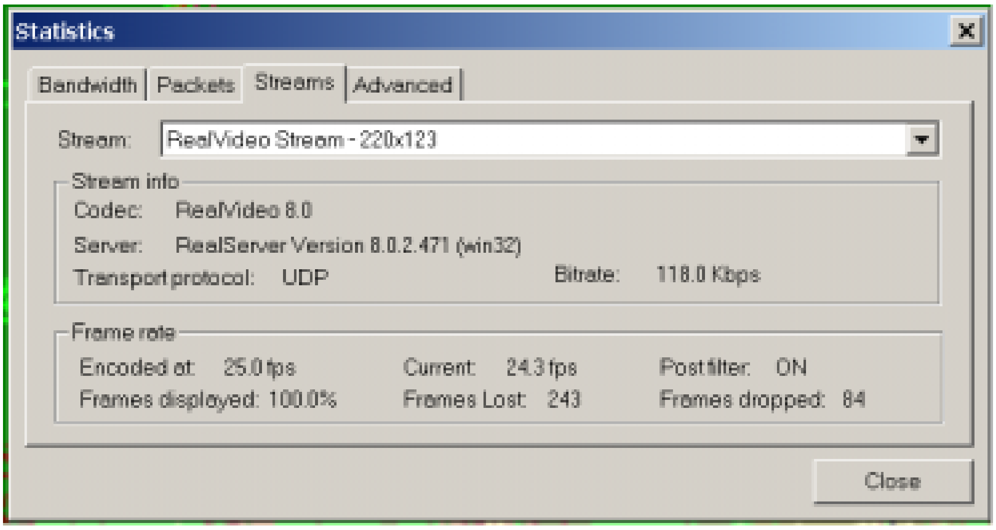

Figure 16 Statistics Tool

Chapter 5

Performance, Evaluation and Analysis

5.1 Explanation of the experiments

The final part of this research was to transmit the same video with two different resolutions (856 x 480 and 220 x 123) to prove that when the bandwidth is narrow it is better to transmit a small resolution video, and when the bandwidth is wide enough it is better to transmit a bigger resolution video. And at the same time if the bandwidth is really small then neither of them will work and both types of video will stop, therefore proving that in those circumstances it is better to transmit a written file with the same information.

This chapter is divided into six parts. The first part explains the two types of files that where used in this experiments. The second part is a set of experiments where 5 students attempted to access the same video file at the same time with better quality and medium quality resolution. The third part of this chapter is a set of experiments where 10 students attempted to access the same video file at the same time with better quality and medium quality resolution. The fourth part of this chapter is a comparison between the previous two sets of experiments. The fifth part consists of analysis and points to consider regarding the previous two parts. Finally the sixth part of this chapter is a set of experiments where students accessed the videos at different times and compared the differences between always sending a video file with a better quality resolution and changing the resolution of the file and also sending a written information of the file, as is recommended in this research.

5.2 Transmitting a video with different quality resolutions

We used two different sets of files in these experiments, the first is a 856 x 480 resolution video file and the second is a 220 x 123 resolution video file. We call the filed 856 x 480 a "better quality resolution" video file and the 220 x 123 a "medium quality resolution" video file.

5.2.1 Transmitting a video with better quality resolution

To do this experiment a file with 856 x 480 resolution was used. Using Real Server at the server end and Real Player at the user end. Using the following URL:

http://150.65.123.82:8080/ramgen/eisuke8-3-2002-856x480.rm

In Real Player a particular file called eisuke8-3-2002-856x480.rm is accessed. Once this file is opened with Real Player Bandwidth Simulator is used as explained in Chapter 3 to simulate different bandwidths and then another tool called Statistics is used to monitor the results of sending this type of video resolution. Information is gathered relating to bit rate and frame per second. Besides this information the amount of time that it took, from the moment a click was made on the link to the moment the video started, is measured. And video quality and audio quality are also taken into consideration.

5.2.2 Transmitting a video with medium quality resolution

A file with 220 x 123 resolution was used to do this experiment. Using Real Server at the server end and Real Player at the user end. Using the following URL:

http://150.65.123.82:8080/ramgen/eisuke8-3-2002-220x123.rm

In Real Player a particular file called eisuke8-3-2002-220x123.rm is accessed. Once this file is opened with Real Player Bandwidth Simulator is again used to simulate different bandwidths and another tool called Statistics is used to monitor the results of sending this type of video resolution. The same information is gathered as in section "5.2 Transmitting a video with better quality resolution". And the results are shown in this section.

5.3 Experiment results with 5 students accessing the same video at the same time

In this second part of this chapter as explained before is a set of experiment done with 5 students accessing a better quality resolution video and a medium quality resolution video at the same time. And then "Initial time to buffer depending on the bandwidth", "video performance", "audio performance", "Bit rate depending on the bandwidth", and finally "frame per seconds depending on the bandwidth" were recorded.

5.3.1 Initial time to buffer depending on the bandwidth

This experiment consisted of recording how long it took for a video to be shown depending on the quality of the video resolution and the bandwidth that was used to watch it. In this research a tool called "bandwidth simulator" was used, see Figure 15 and also "4.4 Other tools used in this research" to learn how to used this tool.

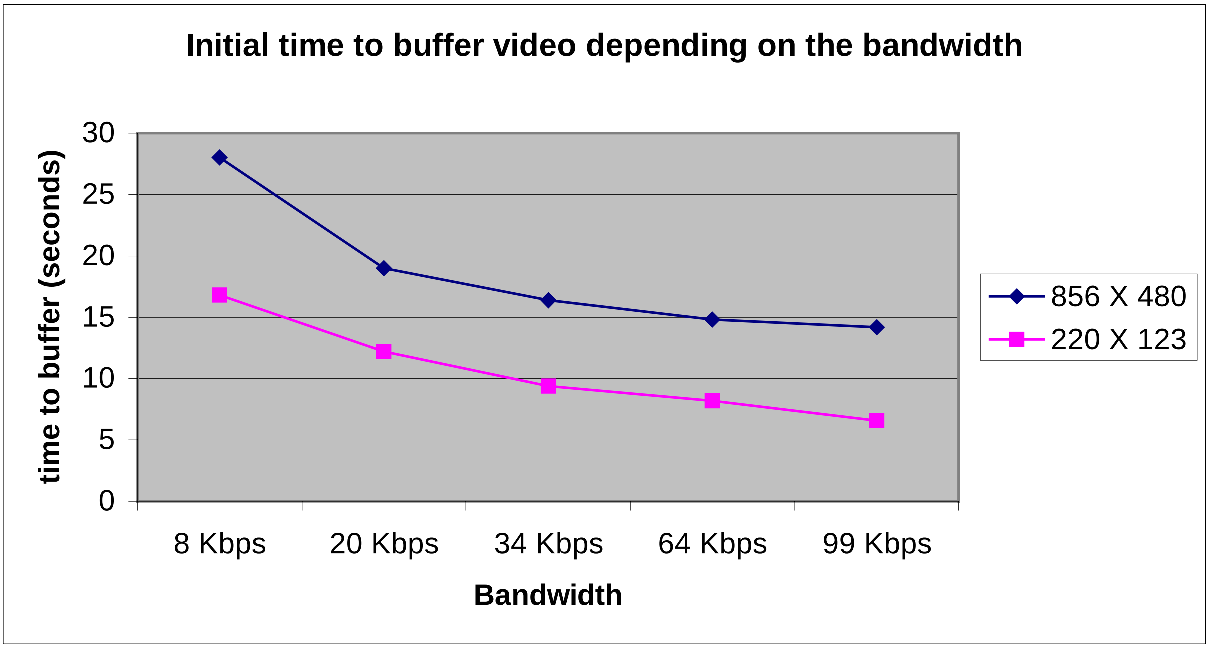

Figure 17 Initial time to buffer video as a function of the bandwidth. (5 students experiment)

As you can see in Figure 17 **Initial time to buffer video as a function of the bandwidth. (5 students experiment)** shows both videos behave in the same way. With the smallest bandwidth 8Kps more time is taken for the videos to be shown, and as the bandwidth is increased less time is taken for the video to buffer. Of course as expected a medium quality video resolution always has a better performance than a better quality video resolution but it is important to notice than when the bandwidth is 99Kps a better quality video takes the same amount of time as a medium quality video does when the bandwidth is 8Kps. Thus is acceptable to send a better quality video when the bandwidth is high 99Kps in order not to sacrifice the quality of the resolution. With these results it is shown that the greater the bandwidth, the better resolution video the user should get, and if the bandwidth is too narrow then a medium quality video should be sent or even text in order for the video not to stall.

5.3.2 Video performance

In this experiment the user was asked to evaluate the quality of the video using the following: “Stopped”, “Stops Sometimes”, “Blocky”, “Fine”. Each category had a rating of 1 to 4 respectively

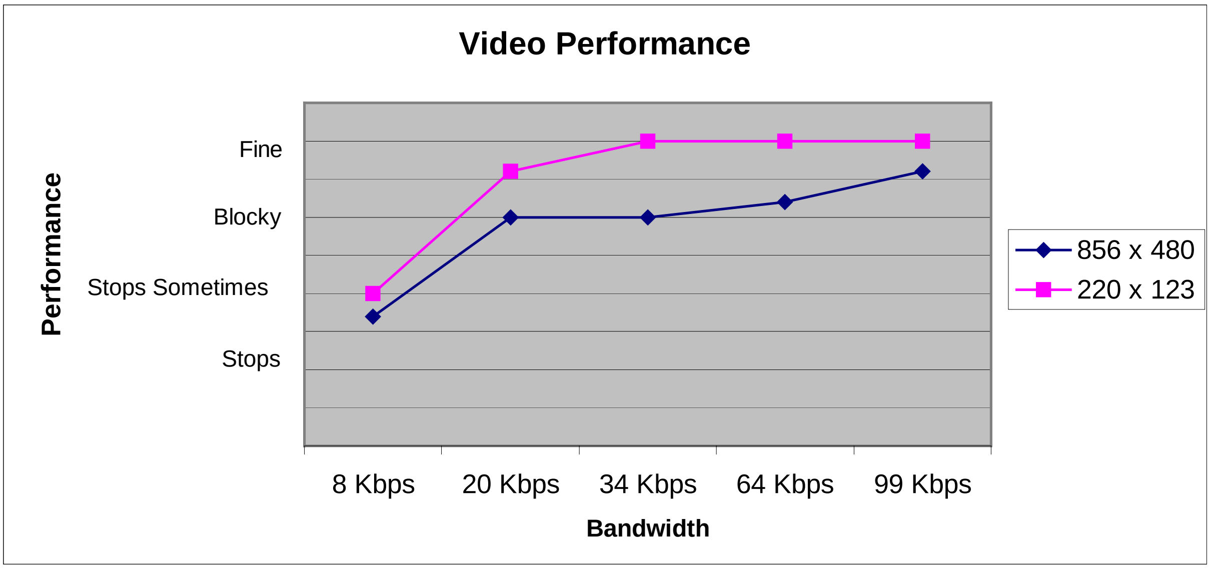

Figure 18 Video performance as a function of the bandwidth used. (5 students experiment)

Figure 18 shows how if the bandwidth is greater (99Kbps) the video has a better performance and if the bandwidth is smaller (8Kbps) then the performance of both videos decreased. Therefore supporting the protocol decision to send written information when the bandwidth is narrow, medium quality video resolution when the bandwidth is a little better (i.e. 34 Kbps or 64 Kbps), and finally sending a better video quality resolution when the bandwidth is greater (99 Kbps). This is because the performance of both videos is almost identical therefore it is better to see a higher quality video.

5.3.3 Audio performance

This experiment consisted of evaluating the audio performance of the two different videos. 5 Students accessed the same video at the same time and evaluated the quality of the sound.

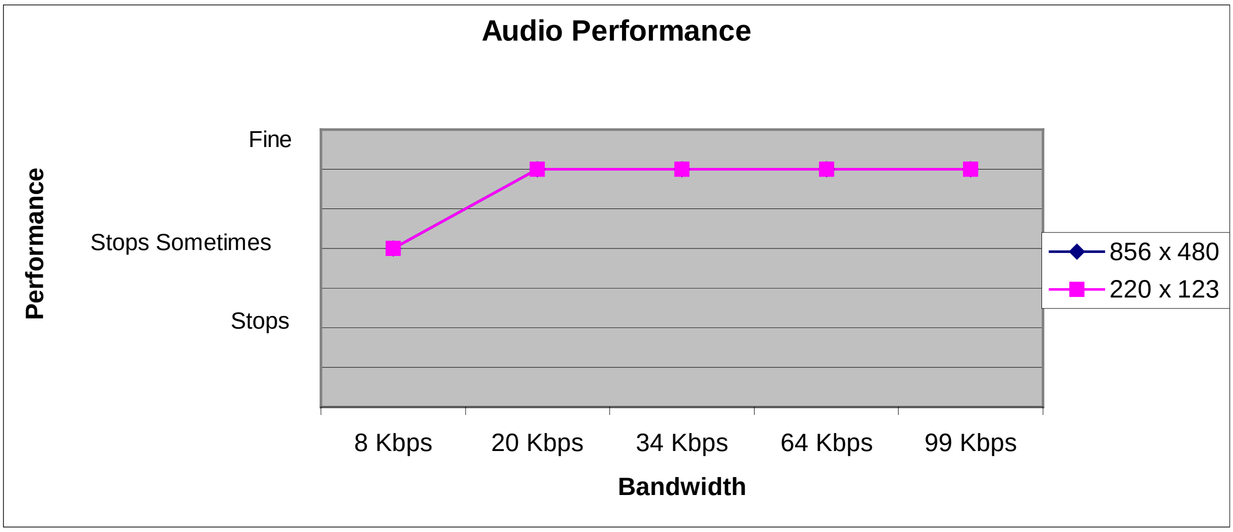

Figure 19 Audio Performance depending on the bandwidth. (5 students experiment)

Not surprisingly the audio performance of both videos were the same at the same bandwidth simulations. See Figure 19

5.3.4 Bit rate depending on the Bandwidth

This experiment was intended to measure the actual bit rate depending on the bandwidth simulator (Figure 15). Once more 5 students accessed the same quality video at the same time and measured the bit rate on each occasion using the tool shown in Figure 16.

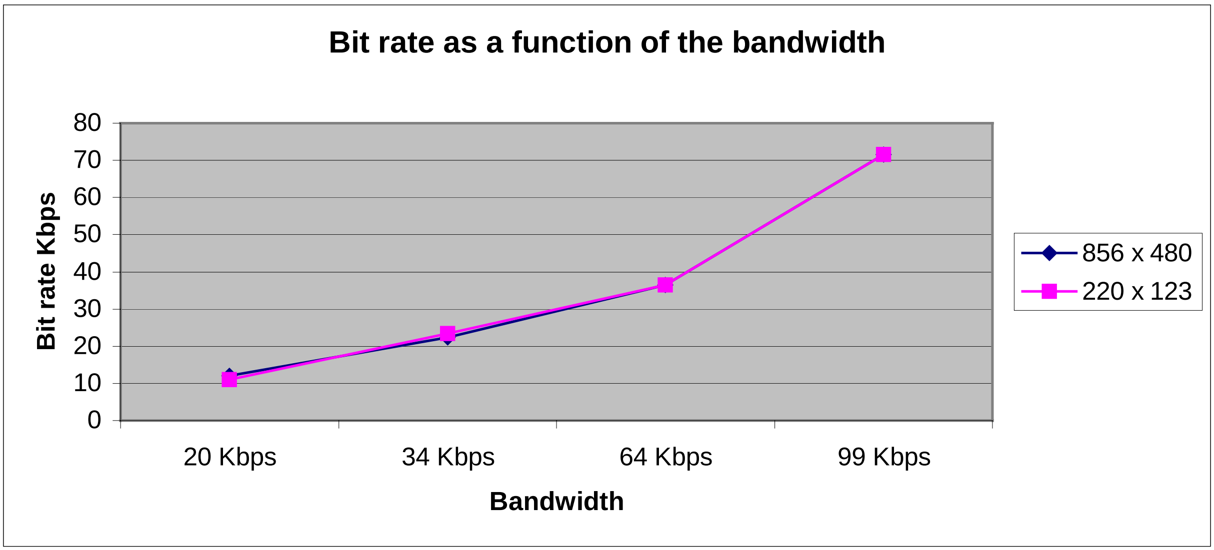

Figure 20 Bit rate as a function of the bandwidth. (5 students experiment)

As expected because the bandwidth was being simulated at a certain bit rate the results of the experiment were that for both videos the bit rate was maintained at a level a little lower than the bit rate specified in the bandwidth simulator (Figure 15). For example when the bandwidth simulated was 20 Kbps then the actual bit rate obtained was always below 20 Kbps. In this case it was closer to 10 Kbps. And the same pattern was repeated for the other simulations.

These results are in accordance with the findings that the greater the bandwidth the higher the bit rate is. Therefore the better the resolution the video transmitted should have. Additionally if the bandwidth is too small then text should be sent, supporting the actions of the protocol Figure 6.

5.3.5 Frame per second depending on the bandwidth

5 students accessed the same video at the same time and measured the frames per second, see Figure 16. As in all the previous experiments bandwidth was simulated, see Figure 15.

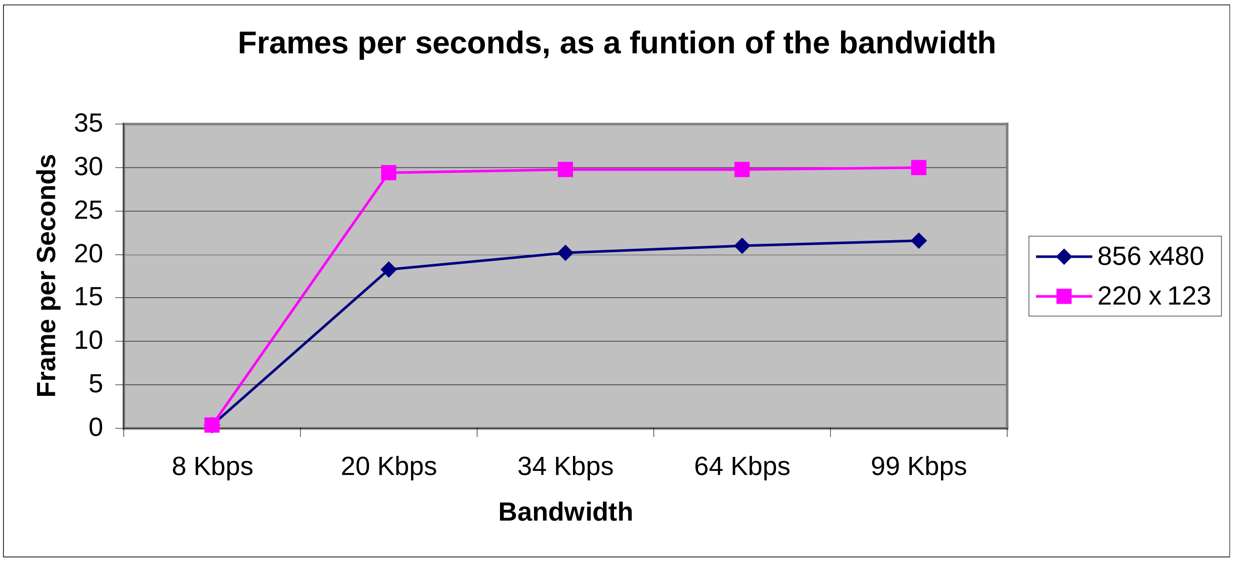

Figure 21 Frames per second as a function of the bandwidth. (5 students experiment)

Figure 21 shows that if the bandwidth is too small (8Kbps) then both videos will stop or send 0 frame per seconds. That is why the new protocol chooses to send text in that case instead of sending any type of video. And as expected a smaller video (220 x123) has a better performance than a bigger one (856 x 480) that is why a smaller video shows more frame per second and a bigger one, that is the difference in Figure 21.

5.4 Experiment results with 10 students accessing the same video at the same time

This section is a set of experiments similar to those in section "5.3 Experiment results with 5 students accessing the same video at the same time" but in this case 10 students were used. In this experiment the bandwidth was changed from 8Kbps to 15 Kbps to see if there were any differences. In this experiment very similar results were obtained as in the previous one.

5.4.1 Initial time taken to buffer depending on the bandwidth

This experiment as in 5.3.1 consisted of recording the time taken to buffer a video depending on the resolution of the video and the bandwidth that it was used.

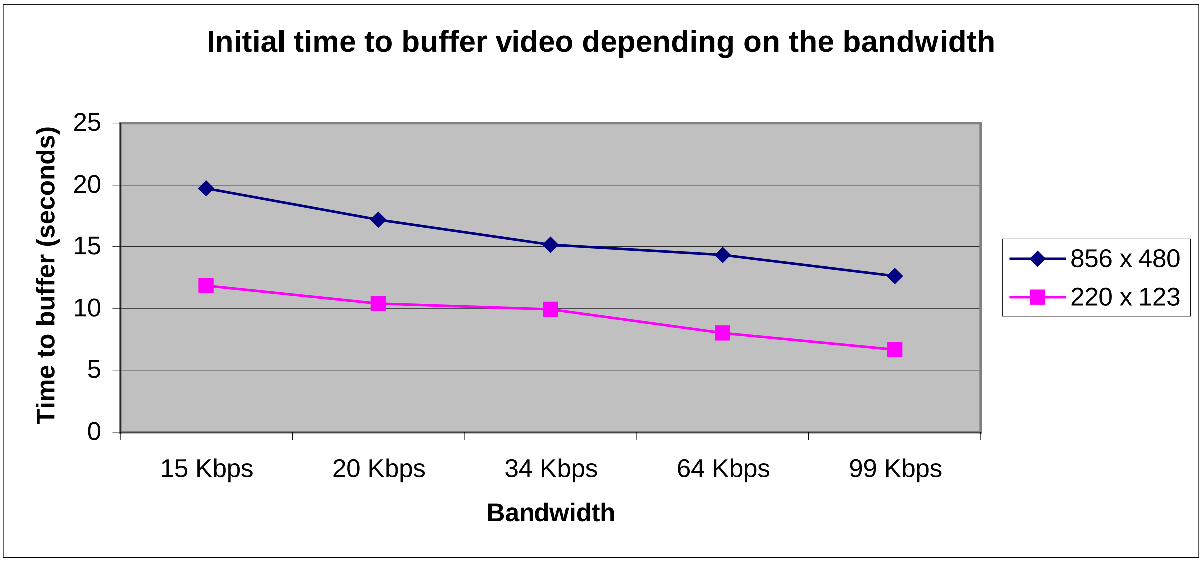

Figure 22 Initial time taken to buffer the video depending on the bandwidth. (10 students experiment)

Figure 22 shows both videos behave in the same way. At the smallest bandwidth (15Kps) more time is taken for the video to be shown, and as the bandwidth is increased up to 99kps the shorter the time the video takes to buffer. Again as expected a medium quality video always gives a better performance than a better quality video but it is noticeable that when the bandwidth is 99Kps a better quality video takes the same amount of time that a medium quality video does at a bandwidth of 15Kps. Thus it is acceptable to send a better quality video when the bandwidth is better (99Kps) in order not to sacrifice the quality of the resolution. With these results it is again reinforced that the greater the bandwidth the better the resolution of video the user should receive. Also if the bandwidth is too narrow then a medium quality video should be sent or even text in order for the information to be successfully obtained.

5.4.2 Video performance

As in section "5.3.2" in this experiment the user was asked to evaluate the quality of the video using "Stopped", "Stopped sometimes", "Blocky", "Fine". Each category had a rating of 1 to 4 respectively

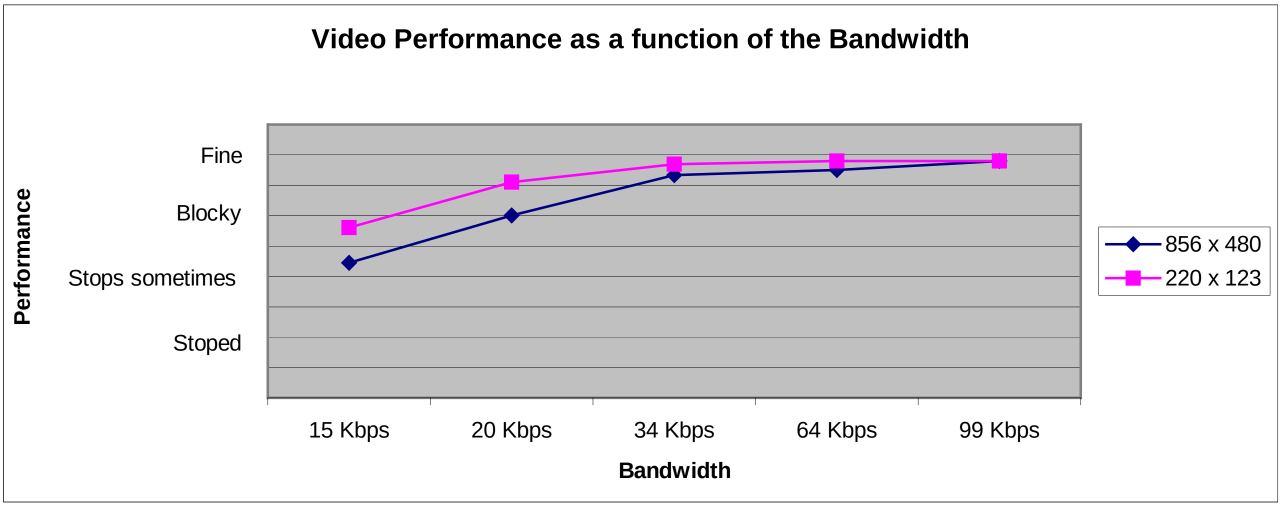

Figure 23 Video performance as a function of the bandwidth used. (10 students experiment)

Figure 23 shows how if the bandwidth is large (for example 99Kbps) the video has a better performance and if the bandwidth is small (for example 15Kbps) then the performance of both videos is decreased. This again supports the decision-making process of the protocol in sending text when the bandwidth is narrow, a medium quality video resolution when the bandwidth is a little better (i.e. 34 Kbps or 64 Kbps), and finally sending a better video quality when the bandwidth is greater (99 Kbps) because the performance of both videos is identical and therefore it is better to see a higher quality video.

5.4.3 Audio performance

In this experiment the audio performance of the two different videos is evaluated. 10 Students accessed the same video at the same time and evaluated the quality of the sound of each video.

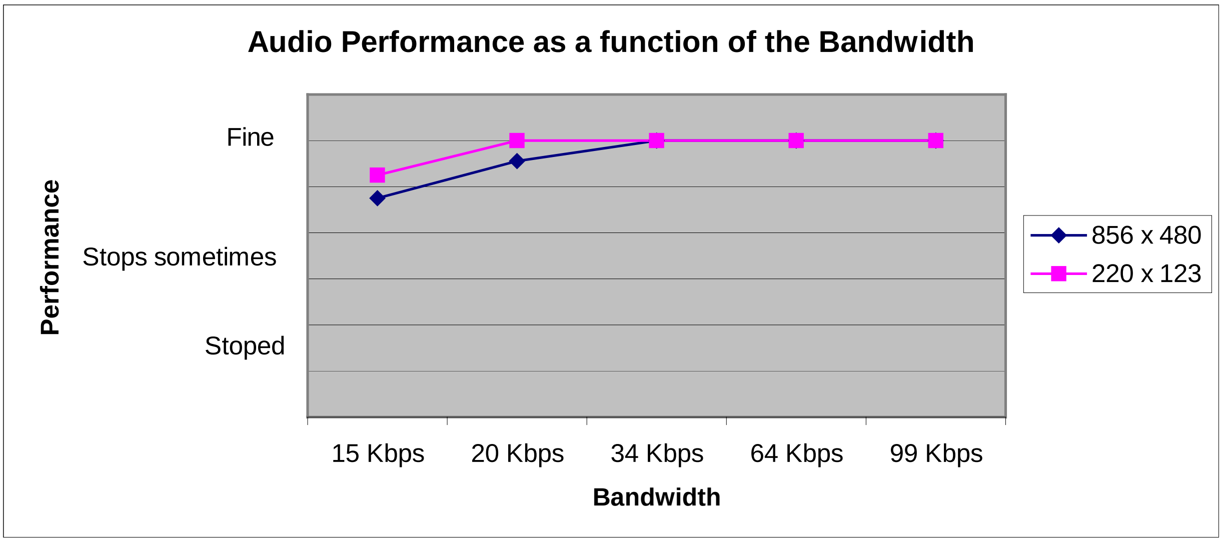

Figure 24 Audio Performance as a function of the bandwidth. (10 students experiment)

As seen in Figure 24 the smaller the bandwidth is the lower the quality of sound, but from 34Kbps upwards the quality of sound for both video resolutions is fine.

5.4.4 Bit rate depending on the Bandwidth

This experiment was intended to measure the actual bit rate depending on the bandwidth simulated (Figure 15). Once more 10 students access at the same quality video at the same time and measure the bit rate on each occasion using the tool shown in Figure 16.

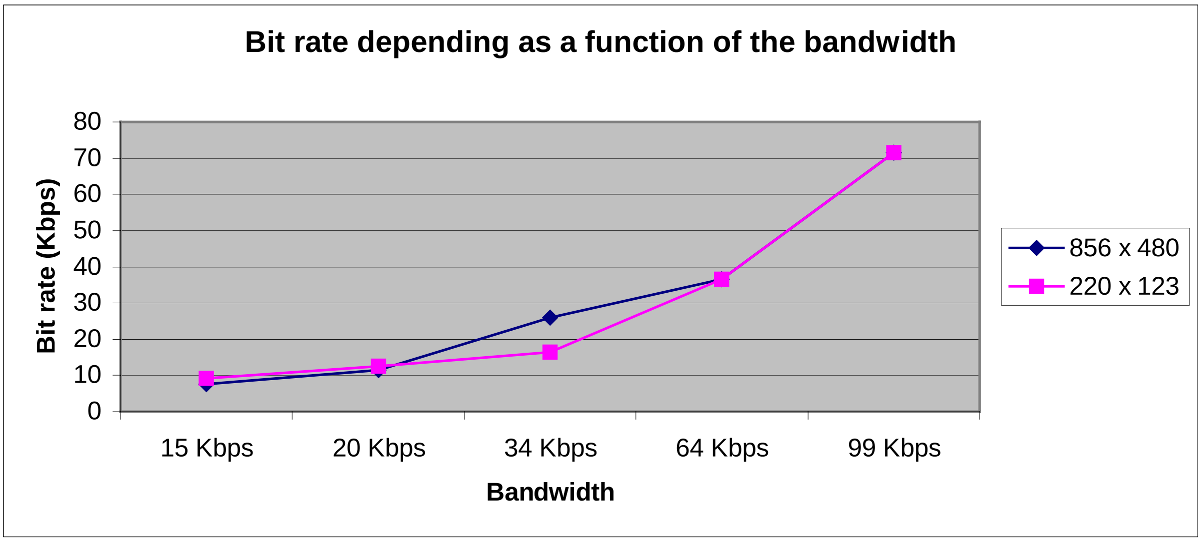

Figure 25 Bit rate as a function of the bandwidth. (10 students experiment)

As expected because the bandwidth was simulated the results of the experiment was that for both videos the bit rate was maintained at a lower level than the bit rate specified by the bandwidth simulator (Figure 15). For example when the bandwidth simulator was asked to simulate 15 Kbps then the actual bit rate obtained was always below 20 Kbps (in this case close to 10 Kbps). Similarly this was the case for the other bandwidth simulations.

These results are in accordance with the theory that the greater the bandwidth the higher the bit rate, and therefore the better resolution the video transmitted should have. Similarly if the bandwidth is too small then text should be sent, supporting what the protocol's actions. See Figure 6.

5.4.5 Frame per second as a function of the Bandwidth

In this experiment 10 students accessed the same video at the same time and measured the frames per second using Statistics. See Figure. As in all the previous experiments the bandwidth was stimulated using "bandwidth simulator" see Figure 15.

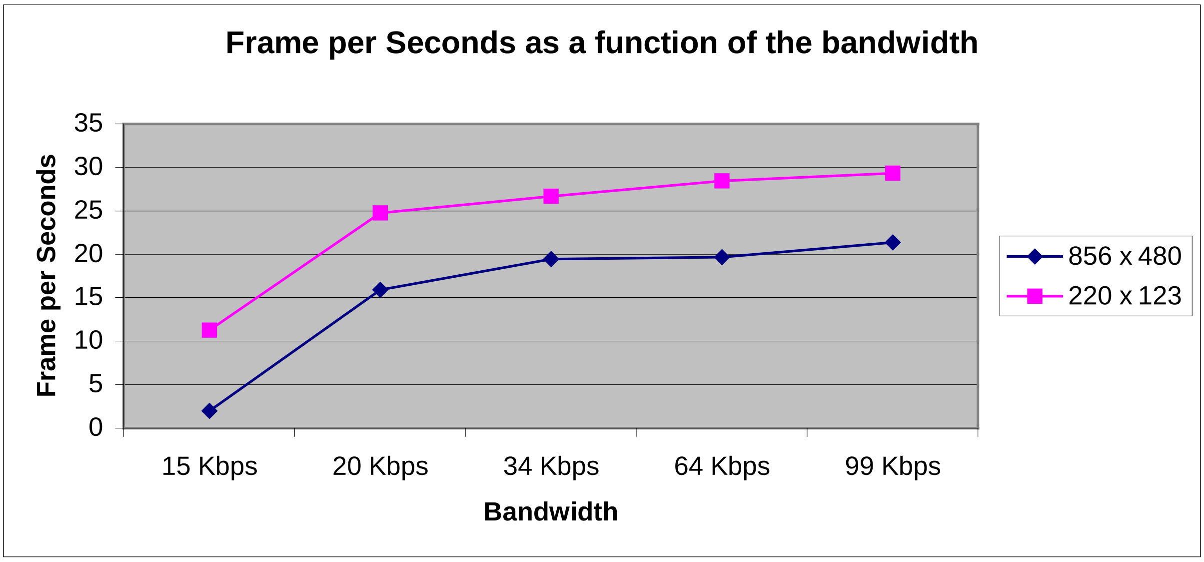

Figure 26 Frames per second as a function of the bandwidth. (10 students experiment)

Figure 26 then both videos will be of poor quality, for example, only 11 frame per seconds in the case of a 220 x 123 resolution video and 0 frame per seconds for a

856 x 480 resolution video. That is why the new protocol chooses to send text in these cases instead of sending any type of video.

5.5 Comparison between the experiment results using 5 and 10 students accessing the same video simultaneously

The fourth part of this chapter is a comparison of the two previous parts "5.3 Experiment results with 5 students accessing the same video at the same time" and "5.4 Experiment results with 10 students accessing the same video at the same time"

The experiments conducted were the same as in the two previous parts, Information was gathered relating time to buffer the video, video and audio quality, bit rate and frame per second.

5.5.1 Initial time to buffer depending on the bandwidth

In this section initial time to buffer depending on the bandwidth is compared between the experiments made with 5 students and 10 students. Once again buffer time means the time taken for a video to be shown and is calculated from the moment a click is made on the link until the time the video appeared on the user's computer depending on the quality of the video resolution and the bandwidth that was used to see it.

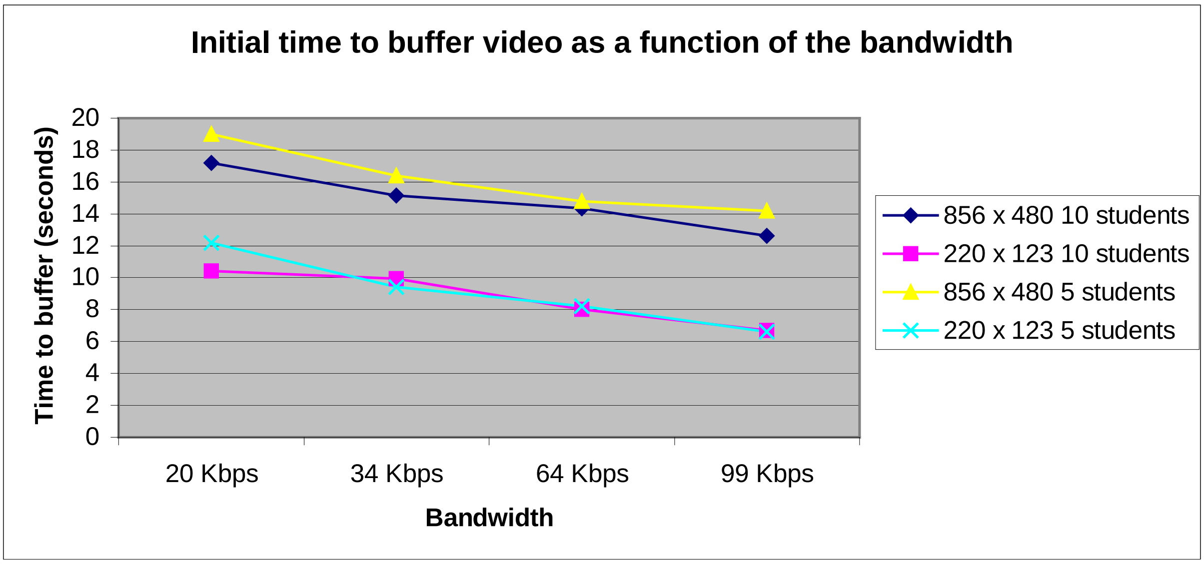

Figure 27 Initial time to buffer video as a function of the bandwidth. Comparison between the experiments made with 5 students and 10 students

As seen in Figure 27 there is little difference between the results using 5 students and 10 students. When a better resolution video was sent it took a longer time to buffer the video in both cases with 5 and 10 students. And when a medium resolution video was sent it took less time in both cases with 5 and 10 students. Therefore there was no difference in using 5 students or 10 students to access the same file and the cause of this phenomenon will be explained in a later section.

5.5.2 Video performance

In this section the video performance was measured using a rating from 1 to 4 where 1 means "the video stopped all the time", 2 means "the video stopped sometimes", 3 is "the video was blocky", 4 is "the video was fine". The results obtained in the previous sections with 5 and 10 students were compared and the results are shown in the following Figure 28.

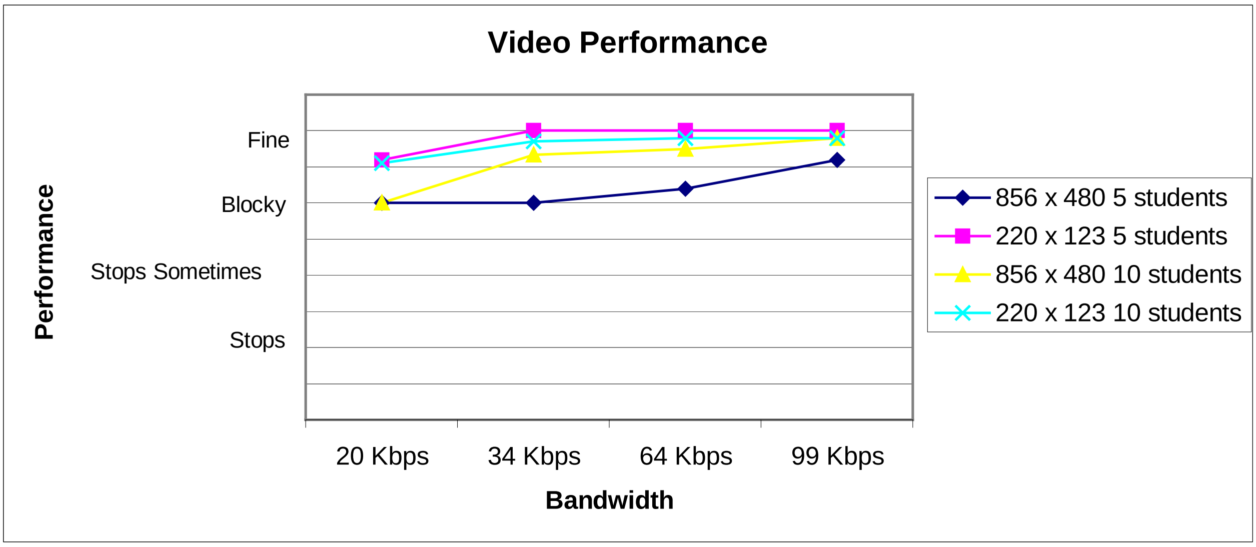

Figure 28 Video performance as a function of the bandwidth. Comparison between the experiments made with 5 students and 10 students

The only real conclusion we can obtained from Figure 28 is that no matter how many students or what the resolution of the video sent the smaller the bandwidth is (20 Kbps) the lower the performance is and the bigger the bandwidth is (99 Kbps) the better the performance of the video. Note that in the 99 Kbps simulation both video resolution with 5 and 10 students are close to satisfactory, therefore backing up the decision of the protocol to send a better quality video.

5.5.3 Audio performance

This experiment measured the audio performance. Again a rating of 1 to 3 is used for the experiment where 1 means "the audio stopped all the time", 2 means "the audio stopped sometime", 3 means "the audio was fine". The results gained in the previous sections with 5 and 10 students were compared and the results are shown in the following Figure 29.

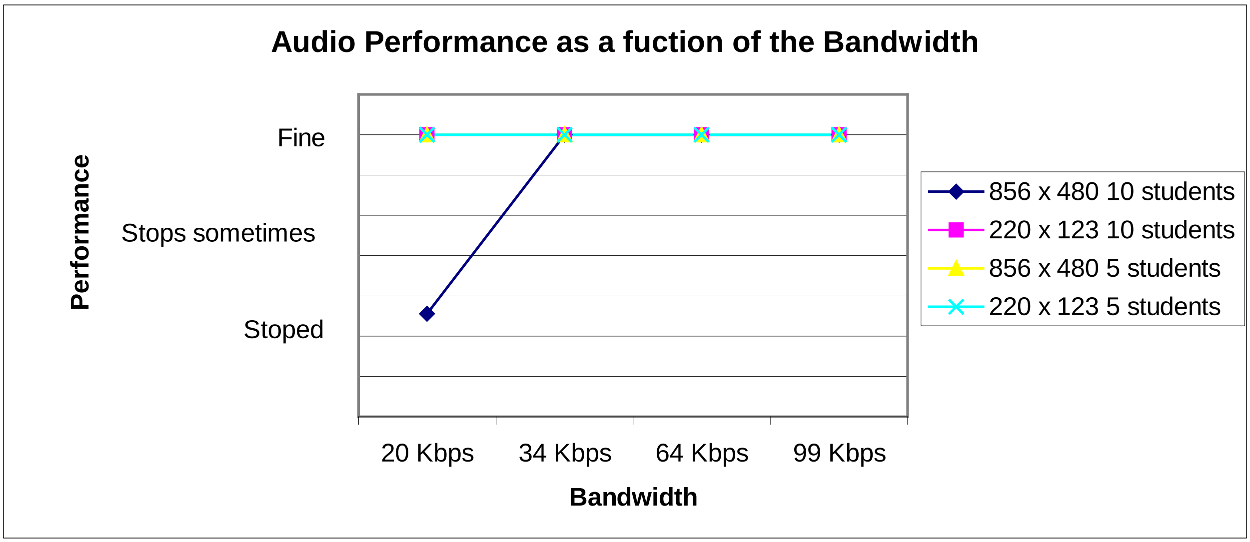

Figure 29 Audio Performance as a function of the bandwidth. Comparison between the experiments made with 5 students and 10 students

As shown in Figure 29 audio is rarely a problem, because no matter how many students or how is the resolution of the video the audio performance is almost always fine. Only in the case where 10 students tried to access the same better resolution video did the audio stop at 20Kbps, thus the protocol would send text instead of a video.

5.5.4 Bit rate depending on the Bandwidth

In this section the actual bit rate was measured depending on the bandwidth using once more bandwidth simulator (Figure 15). The results of these experiments in the previous section with 5 and 10 students are compared. 5 and 10 students in different occasions accessed the same quality video at the same time and measured the bit rate on each occasion using the tool shown in Figure 16.

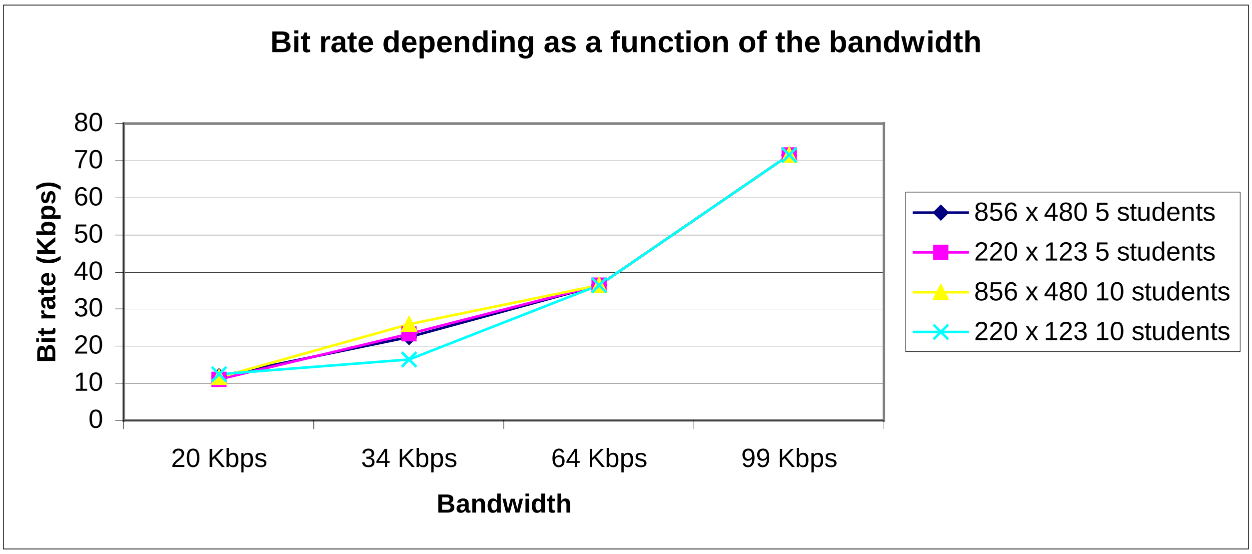

Figure 30 Bit rate as a function of the bandwidth. Comparison between the experiments made with 5 students and 10 students

Figure 30 shows that as expected the bit rate is maintained almost all the time no matter how many students were doing the experiment or the quality resolution of the video because the bit rate was specified by the bandwidth simulator (Figure 15). For example when the bandwidth simulator was asked to simulate 20 Kbps then the actual bit rate obtained was actually always below 20 Kbps. Only when it was 34 Kbps different values were obtained but all of these were under 34 Kbps, ranging from about 15Kbps to 28Kbps. We can attribute this phenomenon to changing conditions on the network.

5.5.5 Frames per second as a function of the Bandwidth

This is the last experiment of this section. The “frames per second depending on the bandwidth using the two qualities of video” is compared using 5 and 10 students in two different time frames. The results are shown in Figure 15.

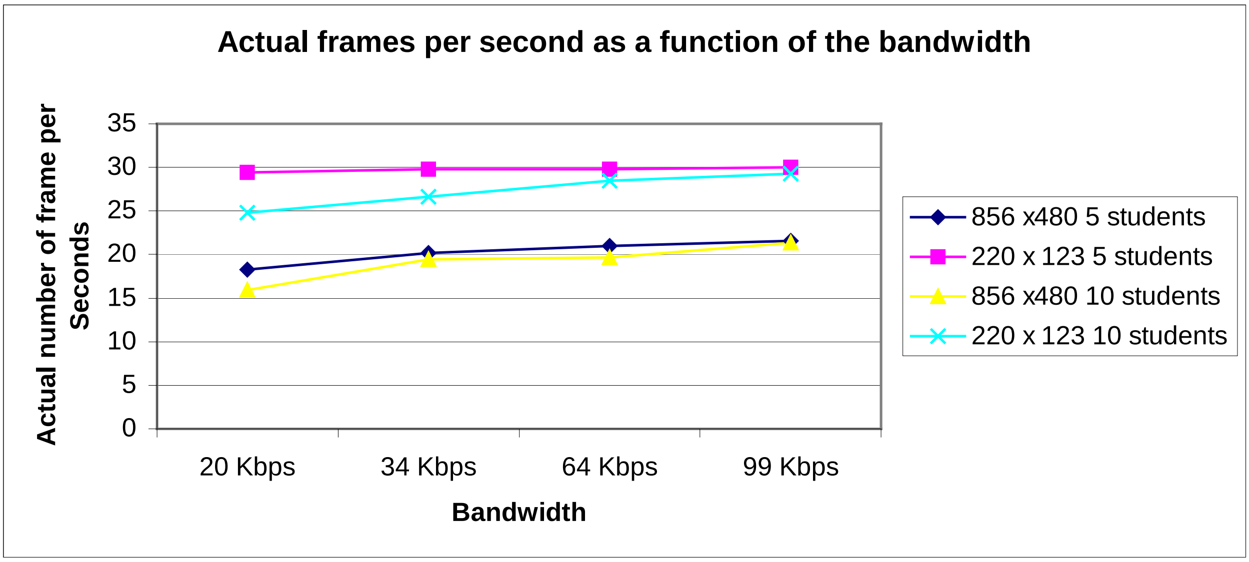

Figure 31 Frames per sencond as a function of the bandwidth. Comparison between the experiments made with 5 students and 10 students

Figure 27 shows that there is little difference between the results obtained using 5 students and 10 students. When a better resolution video was sent the "frames per second" were slower ranging from approximately 16 fps to 20 fps. And when a medium resolution video was sent the "frames per second" increased to a much faster rate ranging from approximately 25 fps to 30fps. Therefore there was no difference in using 5 students or 10 students to access the same file and this will be explained in the next section. See Figure 32.

5.6 Some analysis relevant to the previous section

After comparing the results of the experiments of section "5.3" and section "5.4" the conclusion was that the results obtained were very similar. And after analyzing the data it can be concluded that these results are similar because the bandwidth used is 100mbps using a bandwidth simulator to simulate different bandwidths. So in order to saturate the bandwidth we would have to use the following formula.

$$\text{Maximum amount of students that can access a particular video at the same time} = \frac{\text{Available bandwidth}}{\text{Simulated bandwidth}}$$

Using the previous formula the following table can be constructed. See Table 3

Simulated bandwidth	Available bandwidth/ Simulated Bandwidth	Maximum amount of students that can access the same file
8 Kbps	100 Mbps/8 Kbps	12,500 students
15 Kbps	100 Mbps/15 Kbps	6,666 students
20 Kbps	100 Mbps/20 Kbps	5,000 students
34 Kbps	100 Mbps/34 Kbps	2,941 students
64 Kbps	100 Mbps/64 Kbps	1,562 students
99 Kbps	100 Mbps/99 Kbps	1,010 students
1024 Kbps	100 Mbps/1024 Kbps	97 students

Table 3 Amount of students that can access a particular video at the same time

Because the focus of these experiments has been to measure the performance of the network and not of the server, in the next section the students access the server at different times because the bandwidth is already simulated and the threshold is too high to be reached. Figure 32 outlines this more clearly.

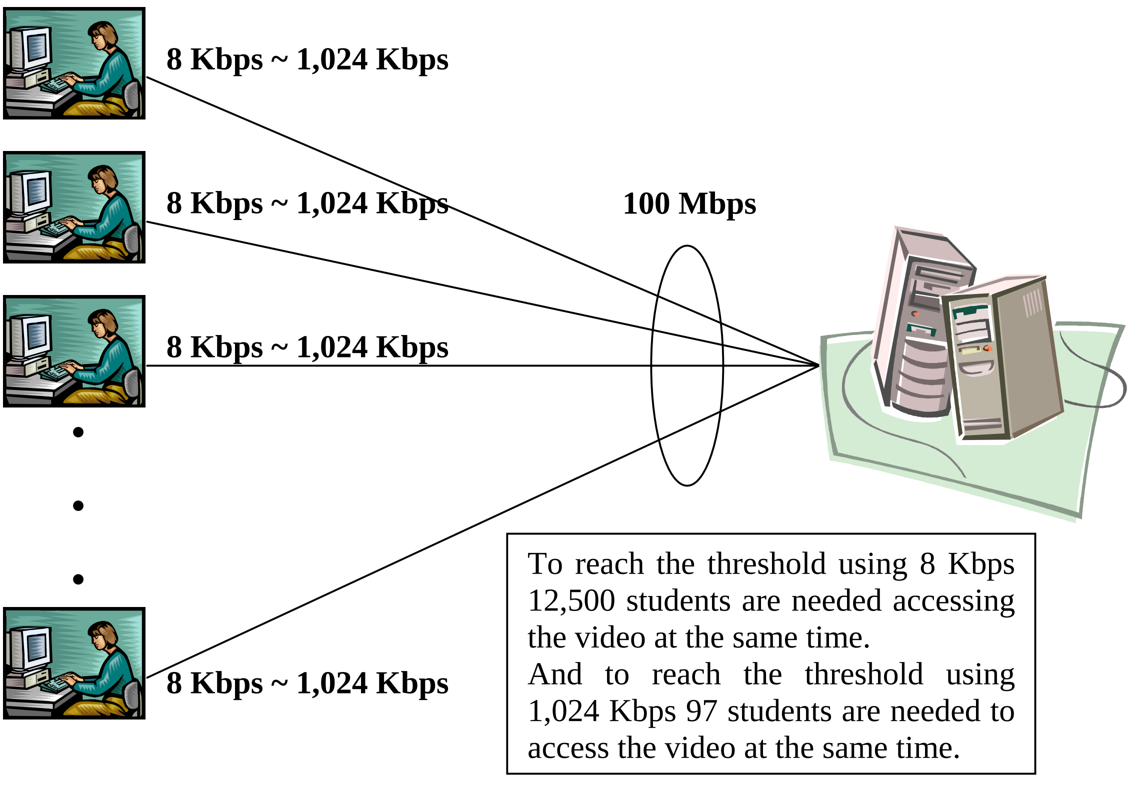

Figure 32 Example of how many students can access a particular video at the same time simulating a 15 Kbps ~ 1,024 Kbps.

5.7 Comparison between the new protocol and previous works

In the sixth and final part of this chapter a set of experiments similar to those made in the previous section are performed using the collaboration of 20 students accessing a better quality resolution video and different type file information depending on the bandwidth simulated. Thus simulating the conduct of the protocol that sends a text file when the bandwidth is small, and sends a medium quality resolution if the bandwidth is average and a better quality resolution video if the bandwidth is very good. Comparing these results to only send a better quality resolution video all the time without taking in consideration the bandwidth. Finally, recording the same information: "Initial time to buffer depending on the bandwidth", "video performance", "audio performance", "Bit rate depending on the bandwidth", and "frames per second depending on the bandwidth".

5.7.1 Initial time to buffer.

In this section the results gathered are compared when sending a better quality resolution video and sending different types of information depending on the bandwidth. Where different types of information could be text or a medium quality resolution video or a better quality resolution video. See Figure 33

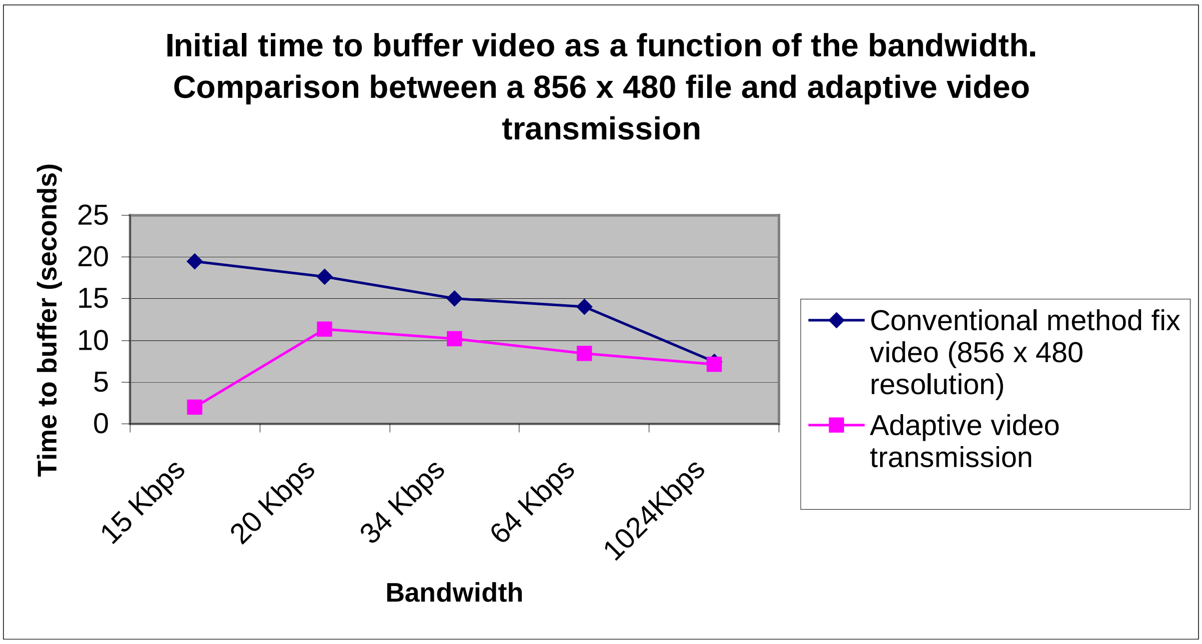

Figure 33 Initial time to buffer video as a function of on the bandwidth.

Comparison between a 856 x 480 file and different types of information

5.7.2 Video performance

In this section the performance of sending a better quality resolution video across all the different bandwidth situations is compared with sending different types of information depending on the bandwidth as the proposed protocol does. In 3.6 See Figure 6.

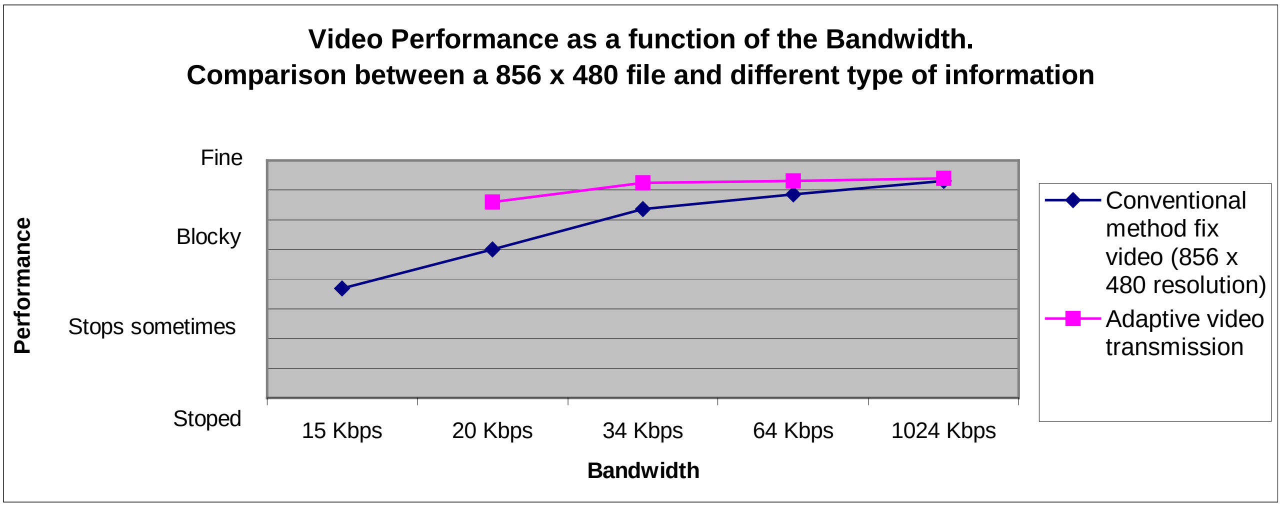

Figure 34 Video Performance as a function of the Bandwidth.

Comparison between a 856 x 480 file and different types of information

Figure 34 shows the performance of sending a video (856 x 480 resolution) across all the possible bandwidths. Also sending different information depending on the bandwidth. For example when the bandwidth is 20 Kbps to 64 Kbps sending a better quality resolution video in one case was compared with letting the protocol choose which video quality to send and it sending a medium quality video. On the other hand when the bandwidth is 1024 Kbps a (856 x 480 resolution) video is compared to how the protocol determines what to send and it chose to send the same video. Finally when the bandwidth is too narrow normally the same (856 x 480 resolution) video would be sent but in this research protocol a text file is sent instead that is why is not possible to compare the video performance.

5.7.3 Audio performance

Similar as in section 5.7.2 the audio performance of sending a video with better resolution quality (846 x 480) is compared to sending different types of information depending on the bandwidth.

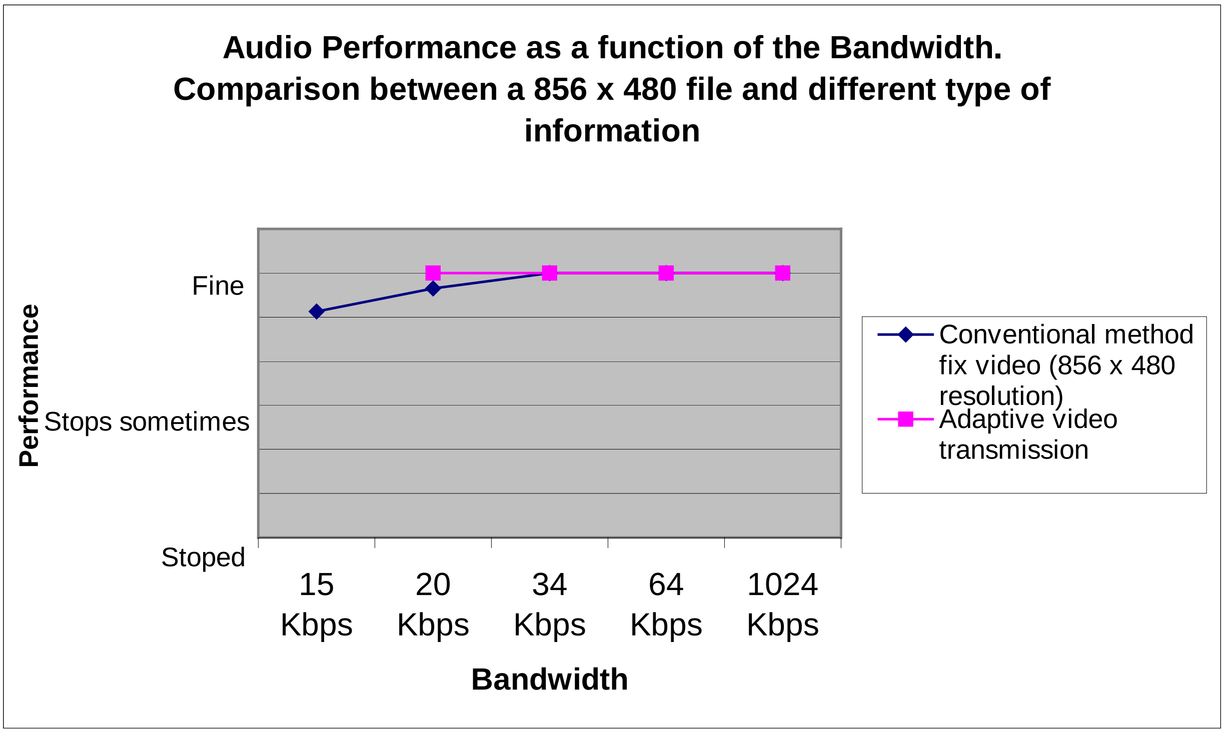

Figure 35 Audio Performance as a function of the Bandwidth.

Comparison between a 856 x 480 file and different types of information

In Figure 35 it is possible to see how at 15 Kbps the difference between audio performance can not be compared because in this research's protocol implementation at that speed a text file is sent instead of a video file. But at the rest of the bandwidths the audio performance of the video sent by the new protocol is satisfactory. On the other hand, if a better quality resolution video is used for all the different bandwidths it is shown that at 15 Kbps and 20 Kbps the audio stops sometimes.

5.7.4 Bit rate depending on the Bandwidth

The bit rate of sending a better quality resolution video across all the different bandwidth conditions and sending different types of information is compared too.

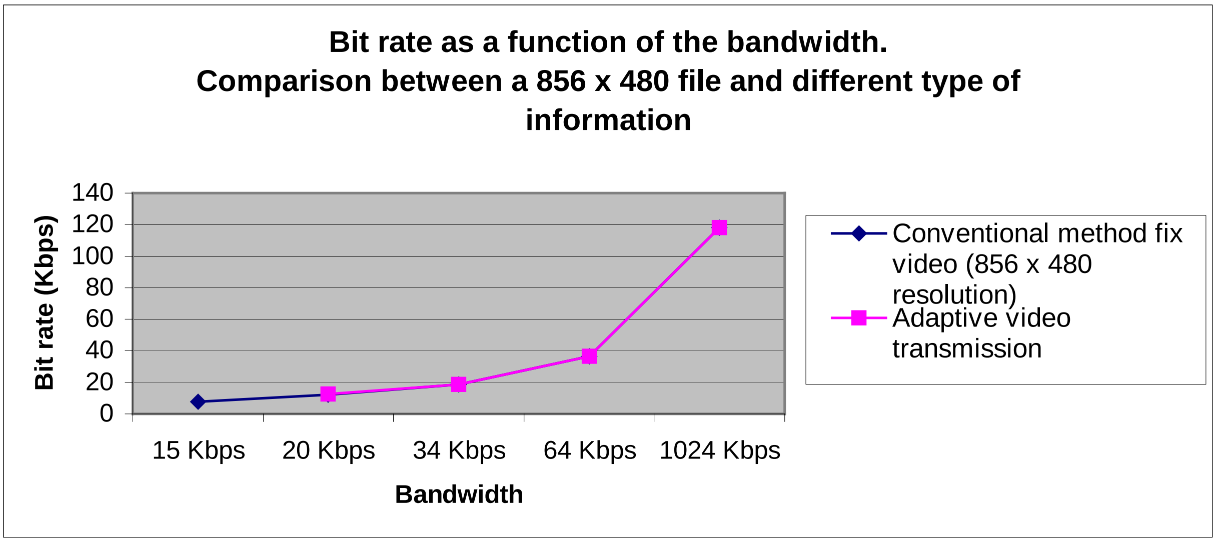

Figure 36 Bit rate as a function of the bandwidth.

Comparison between a 856 x 480 file and different types of information

The results are shown in Figure 36. As explained in section “5.3.4”, “5.4.4” and “5.5.4” the bit rate is maintained the majority of the time regardless of the quality of the video. This is because the bit rate was specified by the bandwidth simulator (Figure 15). For example when the bandwidth simulator was asked to simulate 20 Kbps then the actual bit rate obtained was always below 20 Kbps.

5.7.5 Frames per second depending on the Bandwidth

In this section the frames per second are compared between when a better quality video is sent across all the different bandwidth situations and the frames per second that different types of video quality would present. Of course again when the bandwidth is 15 Kbps the frames per second cannot be compared because the protocol implemented in this research sends a text file instead of a video file.

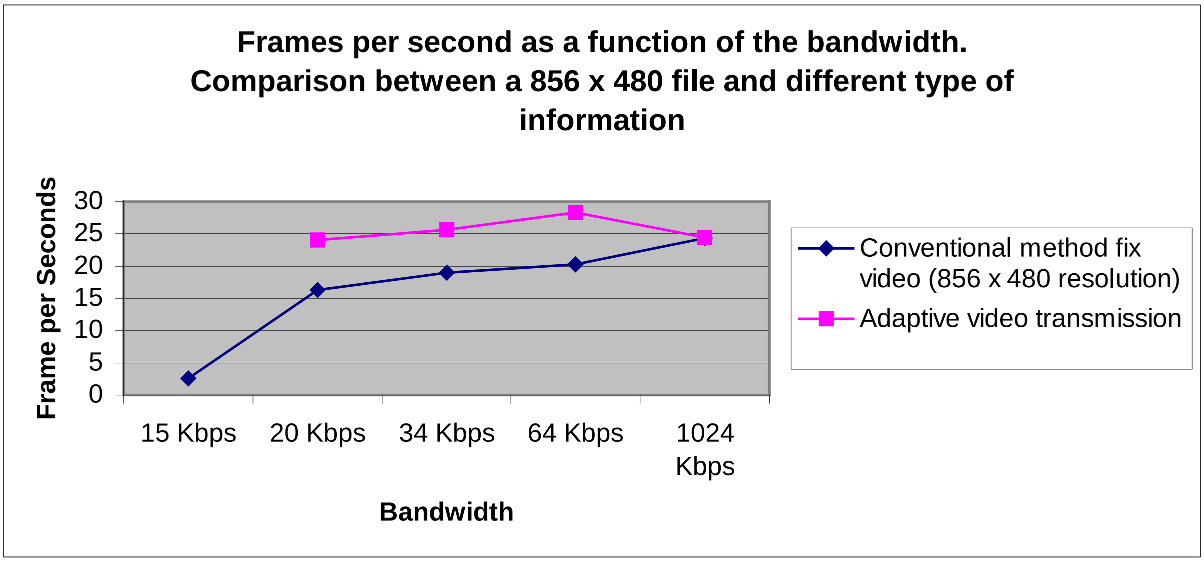

Figure 37 Frames per seconds as a function of the bandwidth.

Comparison between a 856 x 480 file and different types of information

In Figure 37 the line on top shows the behavior of the new protocol. At 15 Kbps the protocol sends a text file which explains why there is no data for the frame per seconds in the bandwidth, then from 20 Kbps to 64 Kbps the protocol sends a medium quality video (220 x 123) which explains the higher frame per second execution and finally at 1024 the frames per second is the same because the protocol sends a better quality video file.

5.7.6 Over all performance

Finally, the last experiment conducted in this research was to ask the students to give us their overall impression of sending the same better quality video across all the bandwidth simulations and sending different information depending on the bandwidth.

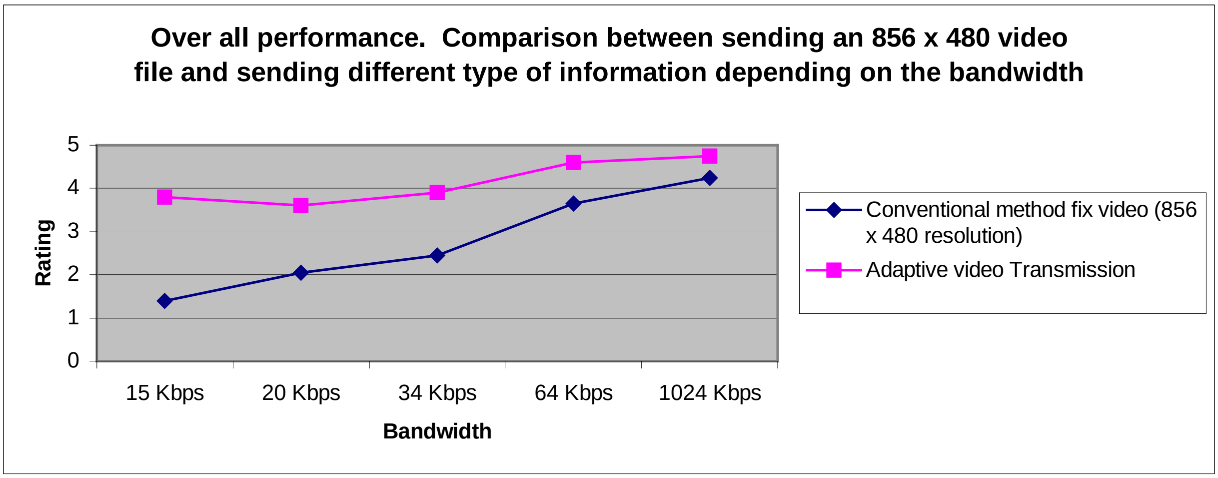

Figure 38 Overall performance. Comparison between sending a 856 x 480 video file and sending different types of information depending on the bandwidth

Figure 38 shows the results of the students' perception of the overall performance of the system. Whereas at 15 Kbps instead of sending a video the protocol sends a text file and between 20 Kbps and 64 Kbps a medium quality video is sent and finally at 1024 Kbps a better quality video is sent. All of this is compared to a situation where a video with a better resolution would be sent all the time.

5.8 Server performance

Althought is another ground of research, this section explain some points of concern about the performance of a web server. There has been much research studying the performance of a web server. Two examples of this are: [29-30].

In the past few years the World Wide Web has experienced phenomenal growth. Not only are millions browsing the Web, but hundreds of new Web sites are added each day [19]. Yet, despite the increasing number of Web (i.e., HTTP) servers in use little is definitively known about their performance characteristics. Web server vendors are quite happy to extol the performance virtues of their products, and industry professionals abound with theories about how to serve data faster; but these virtues and theories are generally based upon anecdotal evidence, gut instinct, or narrow empirical evidence that has little general utility.

Server hardware, Web server software, and a connection to the Internet are required elements of any Web site, and they are all expensive. To generate the best possible performance for any Web site an understanding of the interrelated effects of these three elements on Web server performance is vital.

5.8.1 PCI Bus

Intel defined the PCI bus to ensure that the marketplace would not become crowded with various permutations of local bus architectures implemented in a short-sighted fashion.

Figure 39 illustrates the basic relationship of the PCI, expansion, processor and memory buses. As you can see in Figure 39 the PCI Bus is the connection between the LAN adapter and the CPU, which is the basic route that the video use to be transported from the server to a user in the network.

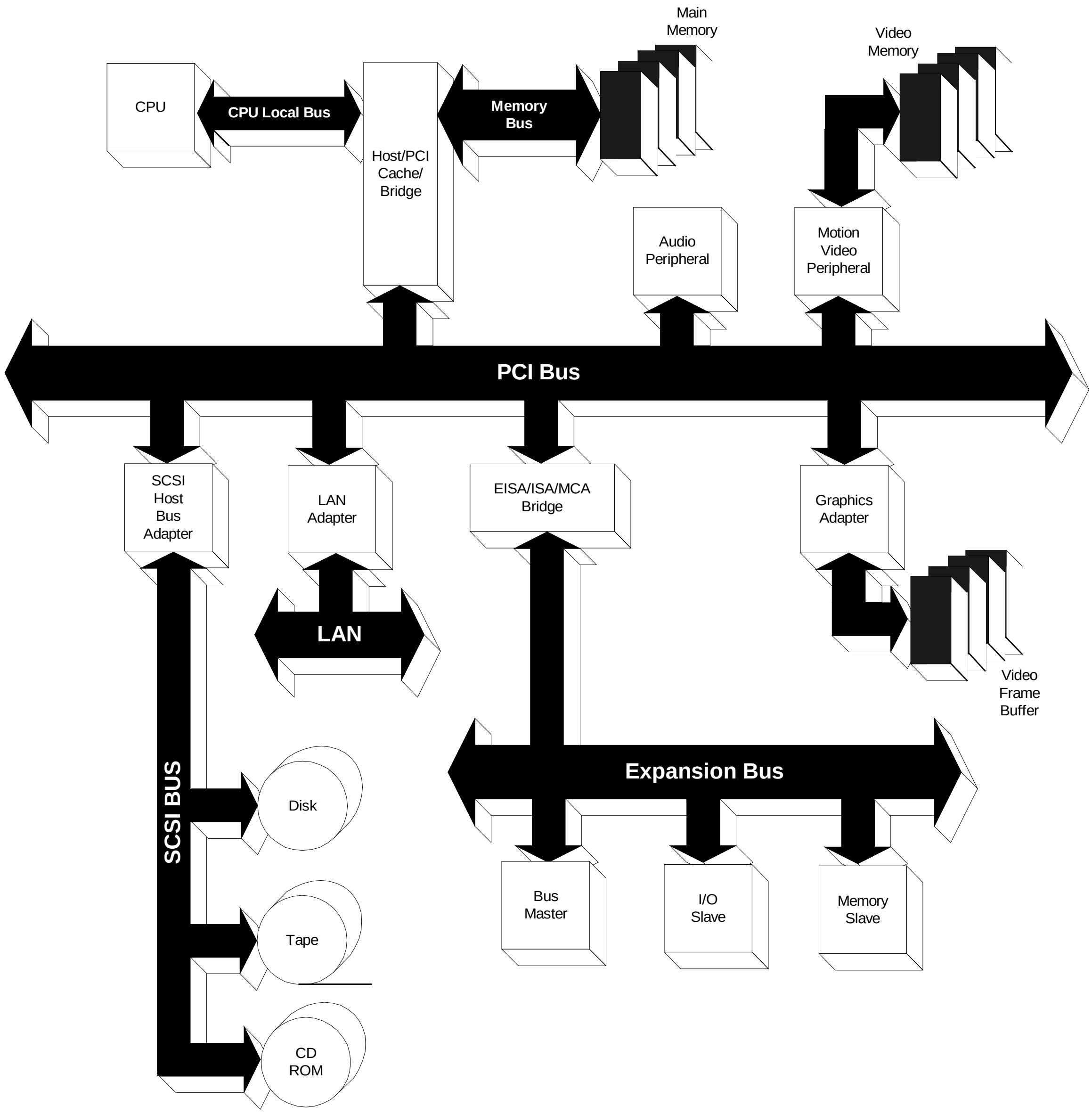

Figure 39 The PCI Bus

The typical PCI devide consists of a complete peripheral adapter encapsulated within an IC package or integrated onto a PCI expansion card. Typical examples would be a network, display or SCSI apater [31].

A normal Pentium III uses a PCI Bus of 32 bits and 33 MHz of speed. Therefore the maximum amount of PCI Bus bandwidth that a pentium is capable of is:

32 bit x 33 MHz = 1,056 Mbps (optimum value used by most companies)

But 1,056 Mbps is not technically correct, because of the different definitions of what “M” stands for. The “M” in “MHz” is 1,000,000 (10^6), but the “M” in “Mbps” is 1,048,567 (2^20). So the band with of the Pentium III is more properly stated as:

(32 x 33)/1,048,567 = 1,007 Mbps (maximum actual PCI Bus size)

But for this research the maximum value that we are going to use for the PCI bus will be 160 Mbps assuming that the other part of the PCI bus is used for other purposes rather that transporting the video information.

In the flowing Table 4 we will present the amount of students that can use this PCI Bus to access videos from the server if we use 160 Mbps as a PCI value.

Simulated Bandwidth for the LAN card	Maximum PCI Bus size/ Simulated Bandwidth for the LAN card	Maximum amount of students that can used the PCI Bus at the same time.
8 Kbps	160 Mbps/8 Kbps	20,000 students
15 Kbps	160 Mbps/15 Kbps	10,667 students
20 Kbps	160 Mbps/20 Kbps	8,000 students
34 Kbps	160 Mbps/34 Kbps	4,706 students
64 Kbps	160 Mbps/64 Kbps	2,500 students
99 Kbps	160 Mbps/99 Kbps	1,616 students
1024 Kbps	160 Mbps/1024 Kbps	156 students

Table 4 PCI Bus capacity

Therefore if Table 4 is compare to Table 3 is posible to see that the server would never become a bottleneck because the speed of the server will always be greater the the speed of the network. For example when 8Kbps is simulated with the bandwith the maximum amount of students that the network can handle is 12,500 and the maximum amount of students that the PCI Bus can handle is 20,000 students. And is the same case in all the other cases, been the speed of the computer always higher that the speed of the network.

5.9 Other points of consideration

5.9.1 Monitoring the network

An experiment consisting in monitoring the network was made. In this experiment we send a ping to 5 different locations over a period of one hour and measure how long it took for the packets to come back. The sites that where monitored, where 2 universities in Panama (UTP, USMA), one server in JAIST (Japan), Toyota (Japan) and a university in United States (University of Illinois).

In Figure 40 we can see the results obtain of monitoring these 5 sites. These results are going to be used for the experiments made from section 5.9.2 thru section 5.9.4. If we analyze Figure 40 we can say that the communication with the server in JAIST (Japan) is very good because the response is close to 0 ms. And that the communication with the server USMA (Panama) is not so good because the response is always bigger that 150 ms. The rest of the communications are average.

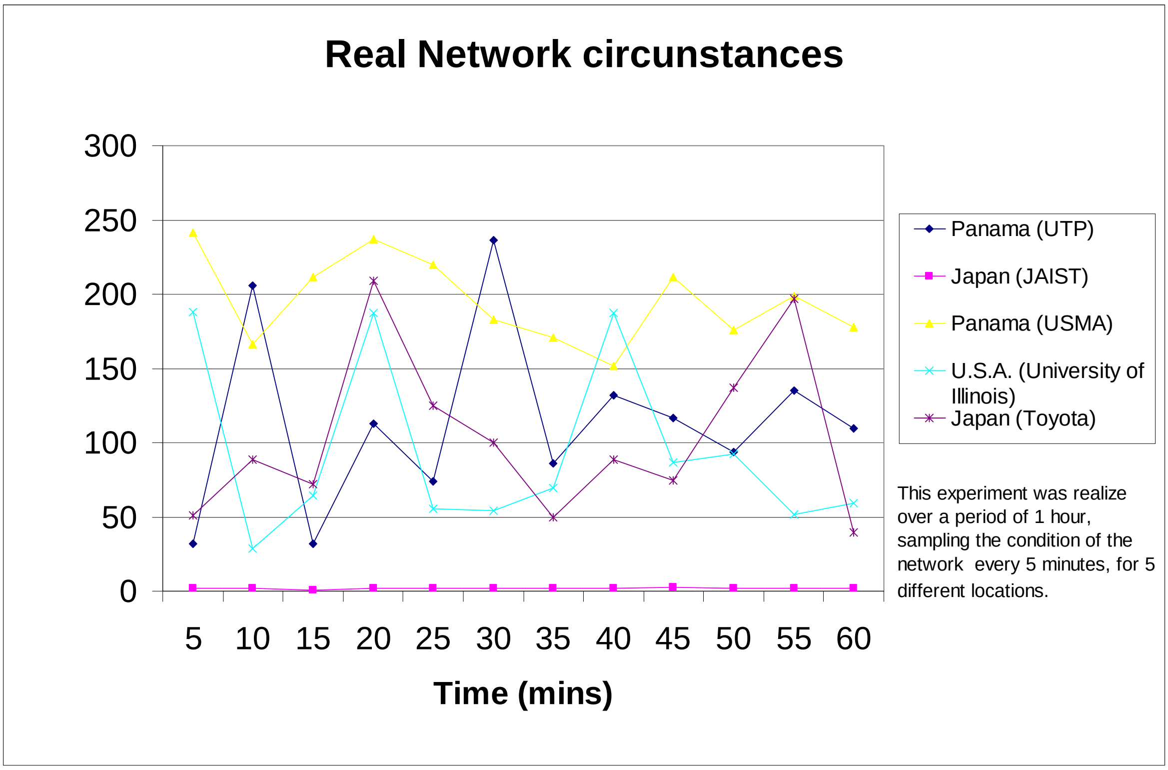

Figure 40 Real Network circunstances

5.9.2 More than 25 frames per second

In this experiment the amount of minutes that a video would transmitt more than 25 frames per second was gathered. 25 frames per second is considered sutable for the human eye to see and smooth video.

In Figure 41 we can see that a small resolution video will always transmit more than 25 frames per second in this conditions on the network. We can also see that only when the communication is real good, as is the case with the server JAIST (Japan), a high resolution video will transmit 25 frames per second over the 60 minutes period that the experiment was conducted. In conclusion we can say that the adaptive size is a better solution when the communication is average because in a period of time of one hour, when the condition of the network is good we can appreciate a high resolution video and the rest of the time a Small resolution video. In previous method you either video a small resolution video all the time of a high resolution that would stall when the communication deteriorate.

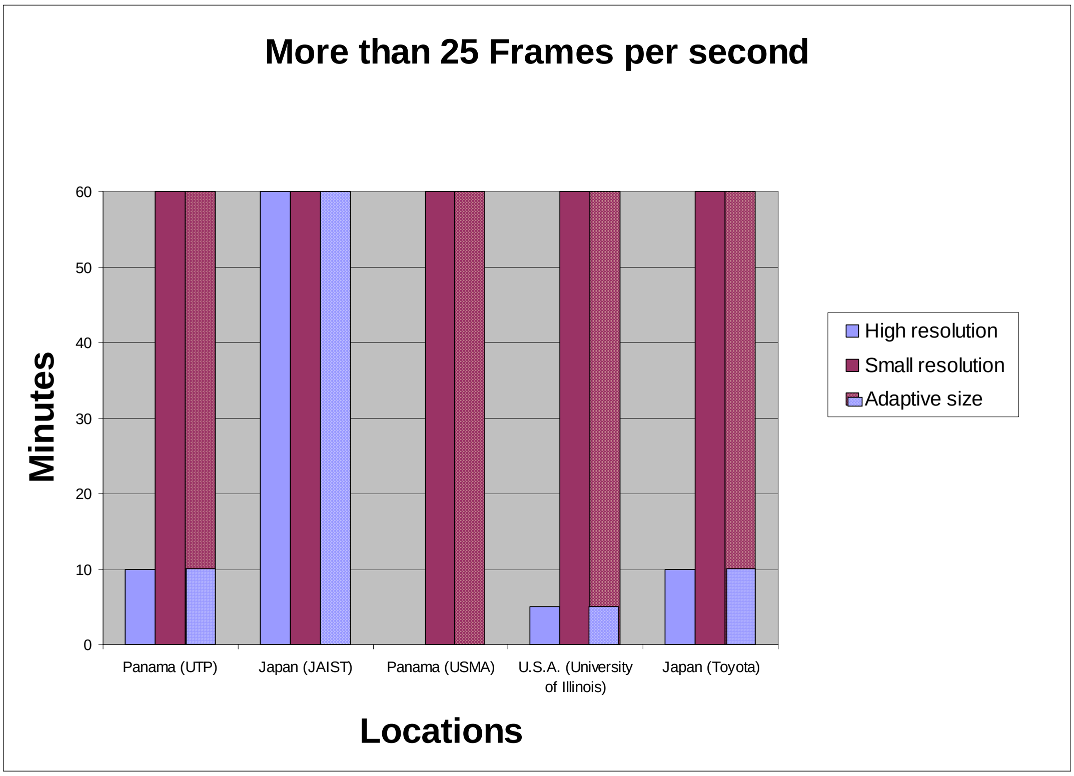

Figure 41 More than 25 frames per second

5.9.3 User's scores

In this experiment we ask 25 students to tell us what was there perception of the video and to rate it as follows: 1 when the video stoped all the time, 2 when the video stoped sometimes, 3 when the video perception was blocky and 4 when the perception of the video was fine.

Figure 42 shows that adaptive size is always perceive in between a small resolution video and a small resolution video. And that the advantage of adaptive size is that it will shift in between video sizes depending on the condition of the network over a period of time, compared to previous method where they either choose a high resolution or a small resolution sacrificing the quality and smoothness of the video.

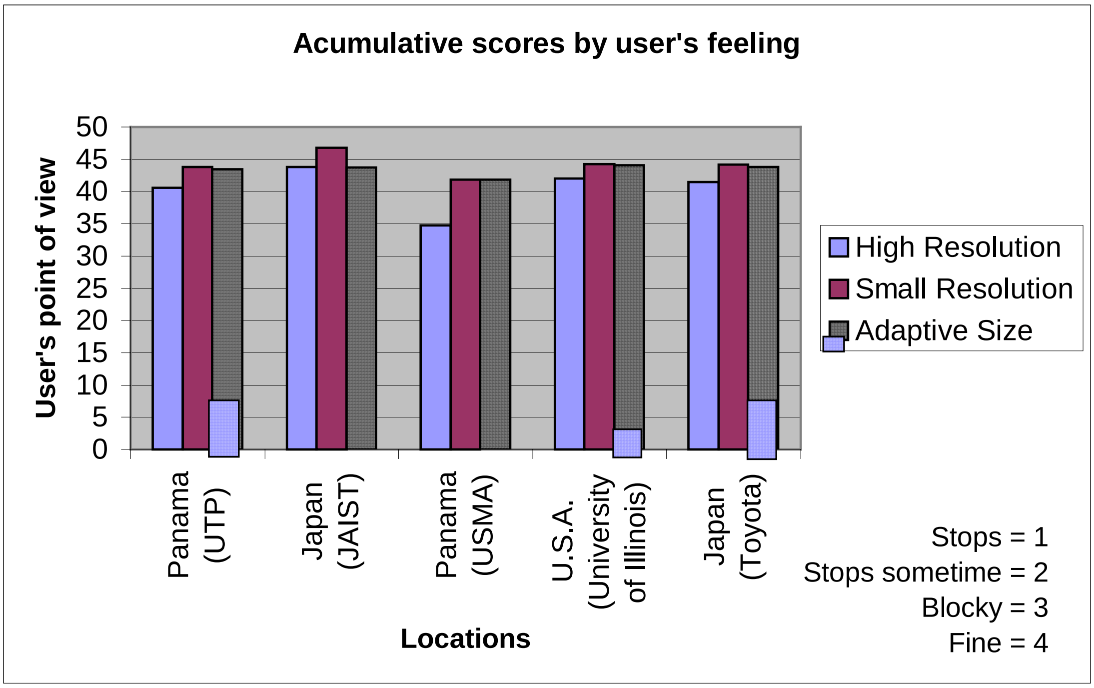

Figure 42 Accumulative scores by user's feeling

5.9.3 Frames per hour

In this experiment the amount frames per hour that a video would transmitt over a period of one hour was messuared every 5 minutes.

In Figure 43 we can see that a small resolution video will always transmit the most frames in an hour in this type of conditions on the network. This graph shows that when the condition of the networks varies then adaptive size will transmit a high resolution video some of the time, therefore the user will get a high resolution when the networks permits it.

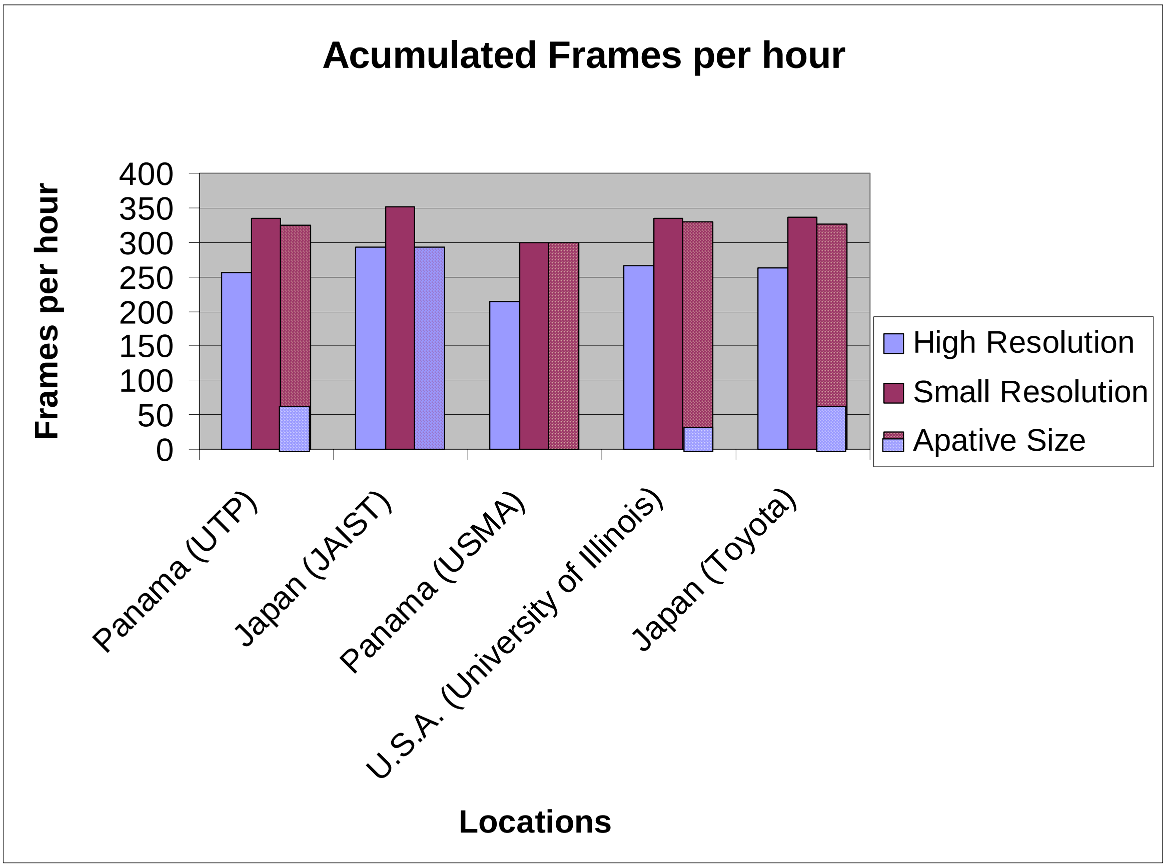

Figure 43 Accumulated frames per hour

Chapter 6

Conclusion

This research has resulted in the successful implementation of a new set of rules that can be called the “Video Transmission For Distance Learning Protocol”.

Chapter 1 of this research provided an introduction to transmission of video over the Internet. It then outlined the objectives of the research and explained the organization of this thesis.

Chapter 2 introduced some of the approaches used in previous work carried out on this topic and critically evaluated this work. Secondly the classification of the methods used to send video over the Internet was explained and then the difference between transmitting whole files or streaming them. Finally an outline of the problem to be solved was provided and a new approach suggested.

Chapter 3 discussed previous protocols and showed that these can be can be classified into three categories: Network-layer protocols such as the Internet (IP), Transport protocols such as User Datagram Protocol (UDP), and Session Control protocol such as Real-Time streaming (RTSP). Next the new protocol was introduced and it was explained that the new system being implemented falls outside all of the three categories because it is an end-to-end solution that runs using all of the mentioned protocols to transmit video over the Internet. The use of ping is then discussed. A detailed explanation of this new protocol is also contained within this chapter.

Chapter 4 relates to the implementation process involved in using the new protocol. This involves using Real Server on the server side and using tools such as Real Player on the user side. Following this, other tools used were mentioned including Bandwidth Simulator and Statistics.

Chapter 5 provides the results of experiments conducted and analysis of the results obtained. These experiments proved that when the bandwidth is narrow it is better to transmit a lower resolution video, and when the bandwidth is wide enough it is better to transmit a video with higher quality of resolution. And additionally if the bandwidth is really small, then as neither of the videos will work satisfactorily and will stop or stall, it is proven that in those circumstances it is better to transmit a text file with the same information. Various experiments were carried out using five or ten students accessing the same video simultaneously to see the impact that a different number of people would have. The initial time to buffer depending on the bandwidth, the video performance, audio performance, bit rate and frames per second depending on the bandwidth were all investigated. A comparison of the two sets of results is also made and an explanation given for the lack of differences found. Finally a comparison was made between transmitting a better quality video and other types of information depending on the bandwidth available.

As discussed earlier in this research there are two modes for transmission of stored video over the Internet: the download mode and the streaming mode. The download mode requires the user to download the entire video file before playback. As outlined the main disadvantage of this full file transfer is that it often takes a very long time to carry out this process. In contrast using the streaming mode it is not necessary to download the whole video before starting playback. The disadvantage of this second method is that due to its real-time nature, video streaming typically has bandwidth, delay and loss requirements. This research is applicable for the streaming mode of transmitting video over the Internet and aims to minimize and circumvent some of the typical problems encountered.

Until now protocols relating to the transmission of video did not consider dynamically changing the resolution of the video to cater for the demands of different bandwidths. Instead the server stored only one video and that was sent to all users regardless of the bandwidth they were using.

The approach used by previous research to deal with the problems caused by narrow bandwidths was just to drop frames from the video files. This resulted in the video stopping, stalling, or being blocky and the overall effect was that the video seen was not very smooth to watch. The minimum frames per second necessary for the human eye to see a video that appears smoothly is 25 frames per second. Below this value the quality will be noticeably affected and this will be detrimental in allowing the information to be processed effectively.

This new protocol combines dropping frames (as in previous work) with a new approach. This new approach takes into consideration the size of the bandwidth available to the user at the time they are requesting the video from the server. The server then dynamically changes the resolution of the video to be shown in response to the bandwidth available. There is also the option for the user to choose to have the video divided into four parts and the bandwidth available determined between each part. This is a response an inherent condition of the Internet: that significant changes in congestion level and hence bandwidth condition can occur over a short space of time.

If the bandwidth available is large then an optimum high quality video will be sent in response to the request. If the bandwidth is moderate, due to some congestion or because the connection to the Internet is not particularly good then a medium quality video with lower resolution will be sent in response to the request. And thirdly if the bandwidth is very narrow due to excessive congestion or a very poor connection to the Internet then no video will be sent because the quality would be too inadequate. In this case a text file providing the contents of the video in written form will be sent instead.

Discussing this in more detail. At present an average good bandwidth obtainable is approximately 100 megabits per second. Assuming that there are no other factors influencing performance then the end users would be expected to be able to view the higher resolution video satisfactorily. At the other extreme if a user using a computer with only a slow modem, for example less than 14.4 Kbps, it will have a narrow bandwidth, and experience excessive congestion on the Internet then in the majority of situations the viewer would only be able to receive the text file though this could be affected by the proximity of the end user and server.

The most valuable attribute of this new protocol is the notion that the end user will always receive the requested information in some form whether it is as a visual communication shown as a video or in the form of a text file. This makes it ideal in the distance-learning context.

As discussed, for distance learning, using the Internet, to be a plausible and efficient method of study the students need to be able to access the information to be studied in a timely manner. The effects of high levels of congestion over the Internet or outdated technology should be minimized wherever possible. This is achieved by the new protocol where video quality is sacrificed, when necessary, in order to achieve the overriding aim which is the provision of information as and when it is requested.

Appendix

The following two forms were used to collect the information shown in chapter 5.

Date:	Student:

856 X 480 Resolution					
Bandwidth Kbps	Time to Show Video	Video Performance	Audio Performance	Bit rate	Frame per seconds
15		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
20		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
34		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
64		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
1024		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		

220 X 123 Resolution					
Bandwidth Kbps	Time to Show Video	Video Performance	Audio Performance	Bit rate	Frame per seconds
15		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
20		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
34		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
64		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		
1024		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine		

Date:	Student:

http://150.65.123.82:8080/ramgen/eisuke8-3-2002-856x480.rm (normal system)						
Bandwidth Kbps	Time to Show	Video Performance	Audio Performance	Bit rate	Frame per seconds	Over all performance of the system
15		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
20		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
34		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
64		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
1024		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5

http://www.jaist.ac.jp/~fabrega/eisuke8-3-2002.htm (my system)						
Bandwidth Kbps	Time to Show	Video Performance	Audio Performance	Bit rate	Frame per seconds	Over all performance of the system
15						1 2 3 4 5
20		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
34		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
64		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5
1024		Stop, Stops sometimes, Blocky, Fine	Stop, Stops sometimes, Fine			1 2 3 4 5

1	Bad
2	Poor
3	Fair
4	Good
5	Excellent

References

[1] Kay Sripanidkulchai, Tsuhan Chen, Network – Adaptive Video Coding and Transmission, Proceedings of SPIE 1999, Vol. 3653, pp. 854 – 861, Visual Communications and Image for Multimedia Engineering, Image Processing Series, CRC press.

[2] Warren Matthews, Les Cottrell - Internet Monitoring in the HEP(High Energy Physics) Community. SLAC-PUB-7961, October 1998.
[3] Warren Matthews, Les Cottrell - Internet End-to-end Performance Monitoring for the High Energy Nuclear and Particle Physics Community. SLAC-PUB-8385, February 2000.

[4] Yun Q. Shi, Huifang Sun - Image and Video Compression for Multimedia Engineering, Fundamentals, Algorithms, and Standards. CRC Press, 2000.

[5] Andrew S. Tanenbaum – Computer Networks. Third Edition. Prentice-Hall International, Inc. 1996.

[6] John Watkinson – MPEG – 2. Focal Press. 1999.

[7] Stephen J. Solari – Digital Video and Audio Compression. McGraw-Hill. 1997. ISBN 0-07-059538-0

[8] David Kerven, Jeff Foust, John Zakour – HTML 3 HOW TO, The Definitive HTML 3 Problem-Solver. Waite Group Press. 1996.

[9] Kamisetty Ramamohan Rao, J. J. Hwang Rao – Techniques & Standards for Image – Video & Audio Coding. Prentice Hall PTR. 1996.

[10] David M. Chandler, Jim Minatel, Bill Kirkner – Running a perfect web site. Que Corporation. 1995.

[11] Larry Wall and Randal L. Schwartz – Programming perl. O'Reilly & Associates, Inc. 1990.

[12] Dapen Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, Jon M. Peha – Streaming Video over the Internet: Approaches and Directions. IEEE Transactions on circuits and Systems for video Technology, Vol. 11, No 1, February 2001.

[13] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson – RTP: a transport protocol for real-time applications. Internet Engineering Task Force, RFC 1889, January, 1996.

[14] H. Schulzrinne, A Rao, R. lanphier – Real time streaming protocol (RTSP). Internet Engineering Task Force, RFC 2326, April 1998.

[15] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg – SIP: sesion initiation protocol. Internet Engineering Task Force, RFC 2543, March 1999.

[16] Shudong Jin, Azer Bestavros – Cache-and-Relay Streaming Media Delivery for Asynchronous Clients. Proceedings of the 4th International Workshop on Networked Group Communication, Boston, MA, October 2002.

[17] Hans Eriksson – Mbone: The multicast backbone. Communications of the ACM 37, 8(1994), pp .54 – 60.

[18] J. Postel - Internet Control Message Protocol. RFC 792, Network Working Group. September 1981.

[19] Wiederspan, J. and C. Shotton - Planning and Managing Web Sites on the Macintosh, Addison-Wesley, 1996.

[20] DETC 2002-2003 Directory of Accredited Institutions. Publish by the Distance Education and Training Council, 2002.

[21] Sandra Kerka - Distance Learning, the Internet, and the World Wide Web. Source: ERIC Clearinghouse on Adult Career and Vocational Education Columbus OH. 1996.

[22] D. V. Eastmond - alone but together: adult distance study through computer conferencing. Cresskill, NJ: Hampton Press, 1995.

[23] A. Bates – W. technology, open learning and distance education. London: Routledge, 1995.

[24] K. Wulf – "Training via the Internet: Where Are We?" training and development 50, no. 5 (May 1996): 50-55.

[25] G. Millbank - Writing multimedia training with integrated simulation. Paper presented at the Writers' Retreat on Interactive Technology and Equipment. Vancouver, BC: The University of British Columbia Continuing Studies. Pp75. 1994.

[26] L. Sherry – Issues in Distance Learning. International Journal of Educational Telecommunications. pp 337-365. 1996.

[27] Noriko Hara, Rob Kling – Students' Distress with a Web-based Distance Education Course: An Ethnographic Study of Participants' Experiences. CSI Working paper. WP 00-01-B1. 2002.

[28] Andrew Feenberg - Distance Learning: Promise or Threat?. National Crosstalk: A Publication of the National Center for Public Policy and Higher Education. 1998.

[29] Louis P. Slothouber – A Model of Web Server Performance, StarNine Technologies, 1996.

[30] Robert E. McGrath – Performance of Several Web Server Platforms, Computing & Communications NCSA, 22 January 1996.

[31] Tom Shanley, Don Anderson – PCI System Architecture, fourth edition. Addison-Wesley, 2000