Chapter 20

INTERNET-BASED DISTANCE LEARNING

Sam Hsu, Nalin Sharda, and Oge Marques

Abstract

Distance learning utilizing the Internet as a transport structure to deliver instructions has been a popular issue in recent years. Education institutions, business circles and government agencies as well are welcoming the Internet era with active efforts to find ways of conducting education, training, and business electronically and cost-effectively. In this chapter, instead of a general introduction to the topic of distance learning, while publications of this nature are widely available, we report three case studies related to Internet-based distance learning conducted in Florida Atlantic University recently. The first case is pertinent to a virtual classroom model that features controlled approaches to avoid unauthorized netsurfing and to maintain an orderly audio conferencing during a class session [1]. The second case reports a collaborative project concerning synchronous distance education between Florida Atlantic University in US and Victoria University in Australia [2]. The third case is related to conducting virtual office hours to enhance the interaction between the instructor and students [3].

1. VIRTUAL CLASSROOMS WITH CONTROLLED APPROACHES

1.1 INTRODUCTION

The purpose of this project was to develop a virtual classroom model that features a controlled approach for synchronizing student activities and regulating group interactions in real-time.

Along with the fast-growing of the Internet we have seen in recent years a booming of virtual classrooms that are taking the advantage of the worldwide networking infrastructure to conduct education electronically. Various virtual classroom systems have been developed so far [4, 5]. Examples of representative virtual classroom systems include Virtual Classroom at New Jersey Institute of Technology [6], Virtual Classroom at Cornell University [7], Saddle Project at Boston University [8], Collaboratories at the US Department of Energy [9], Albatross Project at National Chiao Tung University, Taiwan [10], and Interactive Remote Instruction System at Old Dominion University [11]. Each system may have its own features differing from others. In essence, they all have one common goal – to educate students who are motivated to learn but are blocked by their geographical locations and/or working schedules. However, Two areas that face almost all virtual classroom systems that provide real-time interactivity with students over the Internet but have not been given due attention are a) controlled activities for remote students that are out of the reach of direct supervision from the instructor, and b) coordinated interaction among participants while using audio tools for interactive dialogs. We have therefore developed a virtual classroom model that features a centralized mouse control mechanism to provide the instructor with the capability to call remote students' attention to class and to synchronize class activities, and a Fist-Come First-Serve (FCFS) floor control to regulate participants engaged in audio conferencing in an orderly fashion. The model is named **VCOIN**, acronym for ***V***irtual ***C***lassroom ***O***ver the Internet [12].

1.2 VCOIN FEATURES

The major features of VCOIN include the following:

- Uses the Internet as its networking infrastructure for information distribution.
- Uses existing video/audio conferencing tools for interactive real-time communication.
- Uses standard web browsers for graphical user interfaces.
- Employs a mouse control technique to prevent students from conducting unauthorized surfing in cyberspace.
- Employs a floor control algorithm that allows the instructor centralized floor control while students are allowed to access the floor in an FCFS fashion.
- Introduces the concept of virtual experimenter.
- Provides a chat tool to enable interactive text-based communications between the instructor and the students.

1.3 A VCOIN PROTOTYPE

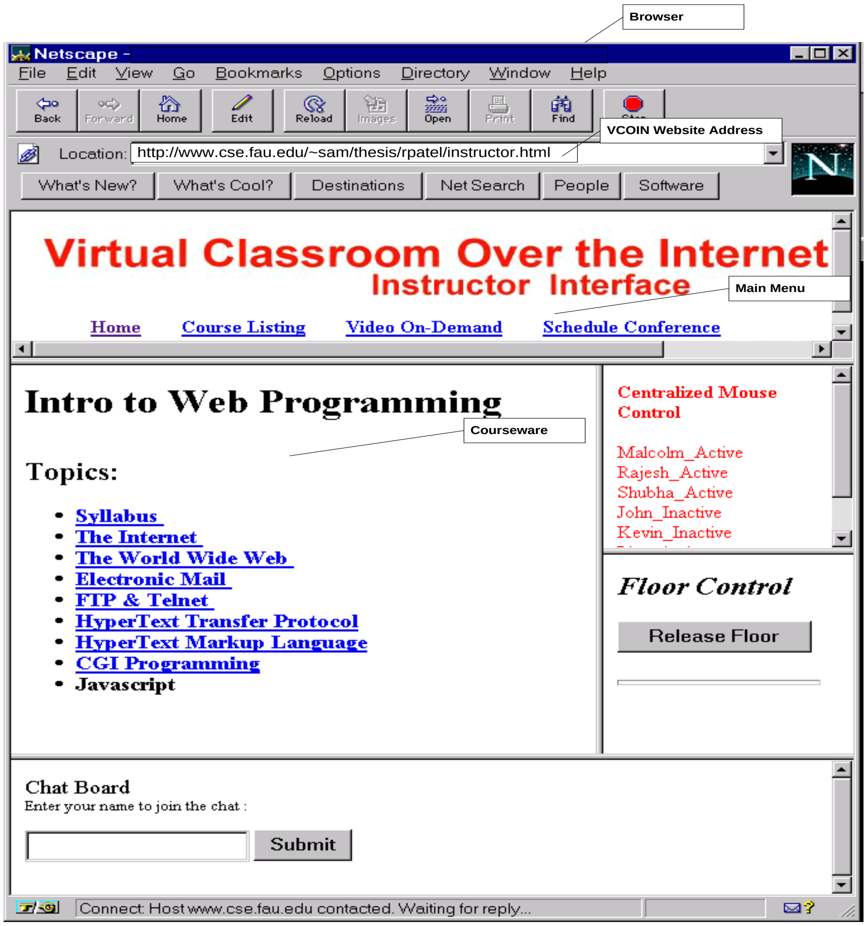
A prototype of VCOIN has been implemented using various programming tools such as HTML, Java, JavaScript, Perl, and C. Several VCOIN trial sessions have been conducted using multiple computer labs that are interconnected via a 100 Mbps Ethernet-type campus backbone. Figure 1 shows a screen capture of the instructor’s interface. VCOIN uses Netscape Navigator for its user interface. Features shown on this screen include a course outline, centralized mouse control, floor control and chat board. Each is housed in a separate frame on the screen.

1.4 CENTRALIZED MOUSE CONTROL USING MOUSE DETECT

In a conventional classroom, the instructor can actually oversee student activities. This situation is quite different in a virtual classroom. A student can be temporarily absent or engaged in netsurfing on a totally unrelated Web site in the case of an Internet-based virtual classroom setting. VCOIN builds into itself a unique feature of checking and controlling the student activities at a remote place. This feature is called *Mouse Detect*. Consider a scenario in which the instructor expects all students to pay attention to what he/she is demonstrating. The instructor will then request the students to place their mouse in a specific area on their screen. VCOIN will detect these activities and notify the instructor of the students’ current status.

Figure 2 shows two screen captures regarding centralized mouse control. The instructor’s interface, Figure 2(a), has a list of participants, shown by student names, and their current status. The status of ‘Active’ indicates that the student has placed his/her mouse in the designated area. This is interpreted as the student is paying attention to the instructor since it prevents him/her from netsurfing the Web or performing other tasks on that computer. When a student has moved his/her mouse pointer outside the specified region, shown in Figure 2(b), the corresponding entry on the instructor’s interface will be changed from ‘Active’ to

'Inactive' immediately. The instructor can then alarm the student (e.g. via an audio tool) for not paying attention to the class.


Browser
Netscape -
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Edit Reload Images Open Print Find Stop
VCOIN Website Address
Location:
What's New? What's Cool? Destinations Net Search People Software
Main Menu
Courseware
Release Floor
Submit

Location: http://www.cse.fau.edu/~sam/thesis/rpatel/instructor.html

Virtual Classroom Over the Internet

Instructor Interface

Home Course Listing Video On-Demand Schedule Conference

Intro to Web Programming

Topics:

- Syllabus
- The Internet
- The World Wide Web
- Electronic Mail
- FTP & Telnet
- HyperText Transfer Protocol
- HyperText Markup Language
- CGI Programming
- Javascript

Centralized Mouse Control

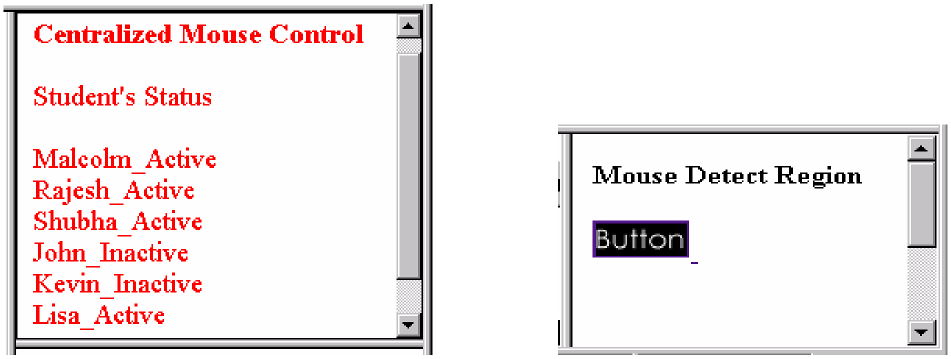
Malcolm_Active
Rajesh_Active
Shubha_Active
John_Inactive
Kevin_Inactive

Floor Control

Chat Board
Enter your name to join the chat :

Connect: Host www.cse.fau.edu contacted. Waiting for reply...

Figure 1. A screen capture of VCOIN instructor interface.



Centralized Mouse Control

Student's Status

Malcolm_Active
Rajesh_Active
Shubha_Active
John_Inactive
Kevin_Inactive
Lisa_Active

Mouse Detect Region

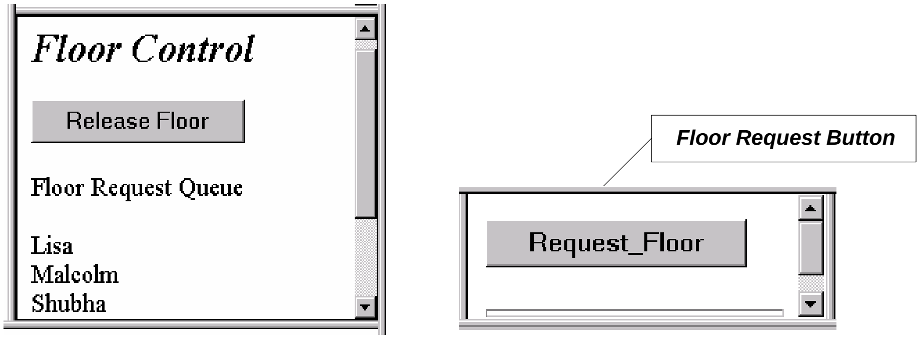
(a) Instructor's frame (b) Student's frame

Figure 2. Screen frames for centralized mouse control.

1.5 FLOOR CONTROL

Floor control is a mechanism that allows the proper use of a shared resource. These shared objects may include video conferencing, audio conferencing, or whiteboard type of applications where more than one participant can have access at a time. Floor control can be implemented using many different algorithms. In VCOIN, a combination of a centralized control (for instructor) and an FCFS (for students) is employed. The instructor always has the authority to speak. A student user is allowed to speak only when he/she is given the permission of floor control by the instructor. This type of control is necessary to avoid chaos and confusion in a conferencing environment and it helps to structure the interaction and reduce the bandwidth requirement.

The floor control algorithm used in VCOIN is an FCFS algorithm for students and centralized control for the instructor. To request the floor control, a student has to submit the request from his/her VCOIN interface. On the other hand, if the instructor would like to release the floor control from a student who currently has it, he/she would have to click on "Release floor" button on the VCOIN interface. The following diagram, Figure 3, shows the two different interfaces.



(a) Instructor's floor control frame (b) Student's floor control frame

Figure 3. Frame layouts of floor control.

The algorithm for implementing floor control is shown in Figure 4. The users who wish to take the floor control submit their request to the controller. The controller puts the request of each user into the FIFO queue. Because the conferencing application is used in a virtual classroom environment, initially the instructor must be having the floor control. Users have three options:

- Request floor control
- Release floor control
- Quit conference

If a user requests for floor control, his/her UserID is added to the FIFO queue only if it is not in the queue. When the user having the floor control requests for release of floor, the controller pops the first element off the FIFO queue and give the floor control to it. The conference application will then allow that user to access the shared resource. If a user request to quit the conference, the floor control mechanism would remove the corresponding UserID from the current user list.

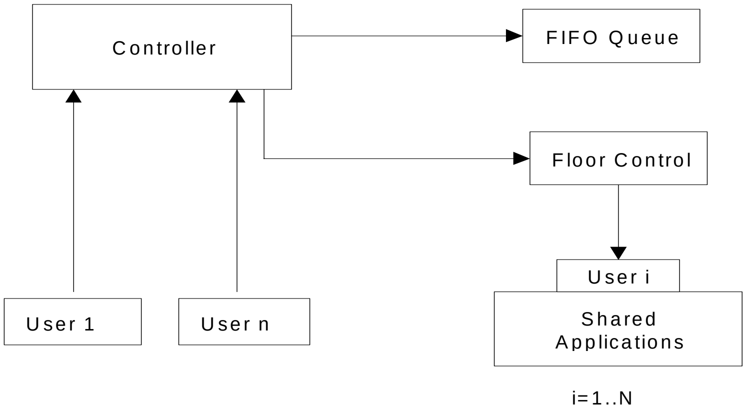
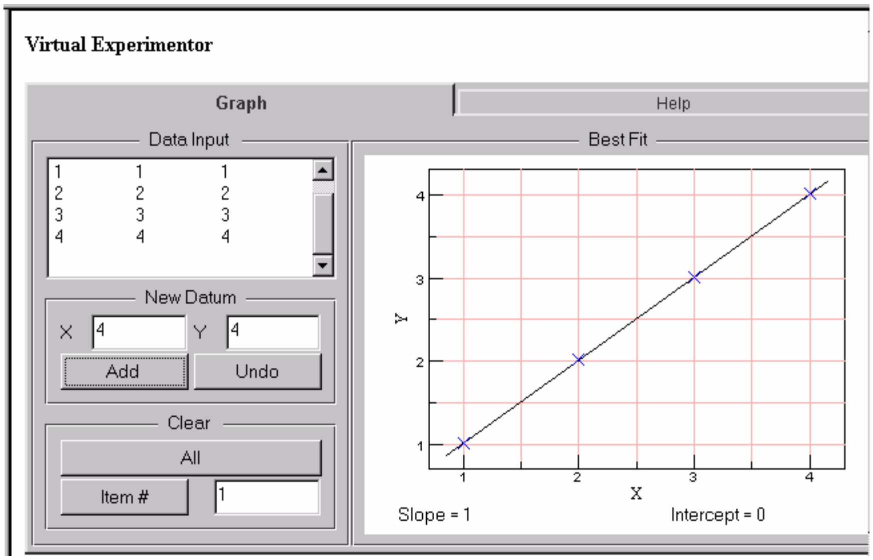


Figure 4. Block diagram of a FIFO floor control.

1.6 VIRTUAL EXPERIMENTER

This feature is used to provide hands-on experience with the material covered in class. It is implemented as a Java applet that can run on any Java-enabled system. Figure 5 shows an example of using it. Figure 6 shows an overview of its implementation



1	1	1
2	2	2
3	3	3
4	4	4

X 4 Y 4

Item # 1

Slope = 1 Intercept = 0

Figure 5. Virtual experimenter in use.

1.7 CONCLUSION

We have discussed various features of VCOIN, a virtual classroom model that can take advantage of a worldwide networking infrastructure of the Internet to conduct education electronically. In addition to having features that are common in most concurrent internet-based virtual classroom systems, we have also focused on two features that are, in general, overlooked in most systems: a) A centralized Mouse Detect feature for monitoring and controlling student activities at a remote place b) A combination of centralized and FCFS floor control for regulation of audio/video communications of student groups. A prototype has been developed. The practicality of the VCOIN model was successfully validated during pilot runs by a group of instructors and students using multiple computer labs. Several screen captures are provided to illustrate the performance of the prototype during these pilot runs.

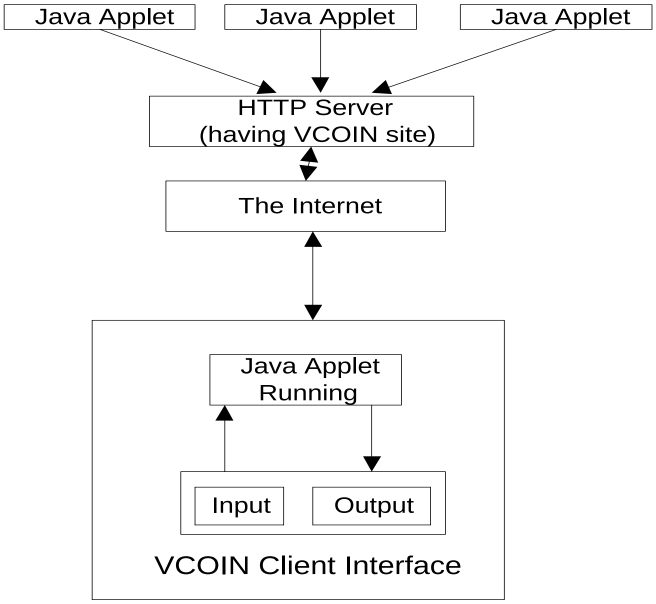


Figure 6. Overview of Virtual Experimenter.

2. VIDEO CONFERENCING EXPERIMENTS BETWEEN FAU AND VU FOR DISTANCE LEARNING USING CLASSPOINT SOFTWARE

2.1 INTRODUCTION

The objective of this project was to investigate the suitability of Internet-based videoconferencing systems to synchronous distance education.

Synchronous distance education uses tools that allow the instructor and the student(s) to log on at the same time, and have virtual face-to-face contact. A simple videoconferencing tool allows transmission of only audio and video information. This is not adequate for distance education. Application-sharing and whiteboard facilities in conjunction with videoconferencing provide a richer environment for distance education. Some of the software tools used for synchronous distance education today include Microsoft's NetMeeting, White Pine’s ClassPoint, and PictureTel's Liveshare. Our aim is to investigate the suitability of Internet-based videoconferencing systems to synchronous distance education. We investigated a number of videoconferencing tools aimed at distance learning. We found the ClassPoint system of White Pine Software [13] one of the most promising tools. We chose this software also because of a special offer made by White Pine to educational institutions in 1998.

Background of the people involved in these experiments is as follows. Nalin Sharda and Sam Hsu are academics at Victoria University (VU) and Florida Atlantic University (FAU) respectively. Satish Gangisetty and Krishna Adusumilli are graduate students at VU and FAU respectively. During the first half of 1998 initial work started at VU and FAU in parallel, almost independently. During the second half of 1998 Nalin Sharda spent a semester at FAU on a visiting position. The inter-university experiments started during this period. Further experiments were conducted in the beginning of 1999.

2.2 CLASSPOINT SOFTWARE

ClassPoint is an instructor-controlled learning environment that can use the Internet for communication and a Web browser as the user interface. It supports multi-point video and audio conferencing, Web tours, class setup and scheduling. It includes shared whiteboard and some other useful features. ClassPoint uses the client-server model [14] shown in Figure 7.

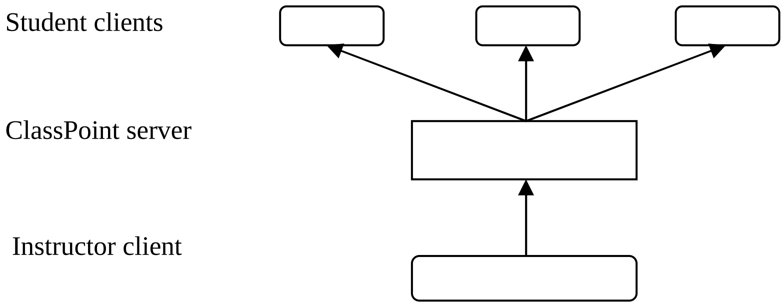


Figure 7. ClassPoint's Client/Server model.

ClassPoint operation is built upon White Pine's CU-SeeMe videoconferencing software and MeetingPoint [15] conference server software. MeetingPoint is the server that provides a complete multimedia conference environment with features such as:

- H.323 compliant audio and video conferencing.
- Data sharing with T.120 standard.
- Security features to control access.
- Billing and tracking support.
- Bandwidth management function.

The main components of the ClassPoint software are:

- Planning Center software.
- Instructor software.
- Student software.

Planning Center Software: The course administrator and the instructor use the ClassPoint Planning Center to create and manage new classrooms. Administrator can create classes and assign students / instructors to these. Each student is assigned a user name and a password. If required, the administrator can also create a group user name and password for general login. It also enables the administrator to keep track of late students. Users enter classes (i.e. login from their workstation) that are currently in session on the ClassPoint Server, using the Planning Center. When a user enters a class the Class Point Student or Instructor application is launched and connected to the specified class. Once a user is connected to the class, (s)he can have a multimedia conference with other users in the same class. ClassPoint Planning Center software requires Windows NT workstation version 4.0 or server with Microsoft Service Pack 3 with a 166 MHz Pentium processor. A minimum of 64 MB RAM and 23 MB of hard disk space is required. TCP/IP networking software and MeetingPoint Conference Server software version 3.051 are required.

Instructor Software: ClassPoint Instructor is the client component through which the instructors can login from a remote site and conduct a class. The instructor manages student participation and interaction in the classroom. The instructor has full control over the students' privilege to participate in the class. The instructor can communicate with the students using audio and/or video, chat, and whiteboard [16,17,18] tool. The instructor also has the ability to conduct Web tour(s) for the class.

Student Software: ClassPoint Student is the client component through which the students can join and participate in the online classes. With the help of this software, students can view class materials before the class or after the class through the ClassPoint server. A student can request the instructor to grant spotlight. Once the spotlight is granted, the entire class can see and hear the student with the spotlight. A student can conduct public or private chat with other students or the instructor.

System Requirements: ClassPoint Instructor and Student require a minimum of 100 MHz Pentium processor with Windows 95 or Windows NT 4.0. The RAM should be at least 32 MB and the hard disk space of 12 MB is required. TCP/IP network connection, video camera, and headphones are essential. The speed of the Network connection should be at least 33.6 Kbps.

2.3 EXPERIMENTS SETUP

We conducted a series of distance education experiments between Victoria University (VU) and Florida Atlantic University (FAU). Figure 8 depicts the logical connection between the various clients and the server in the ClassPoint / MeetingPoint system.

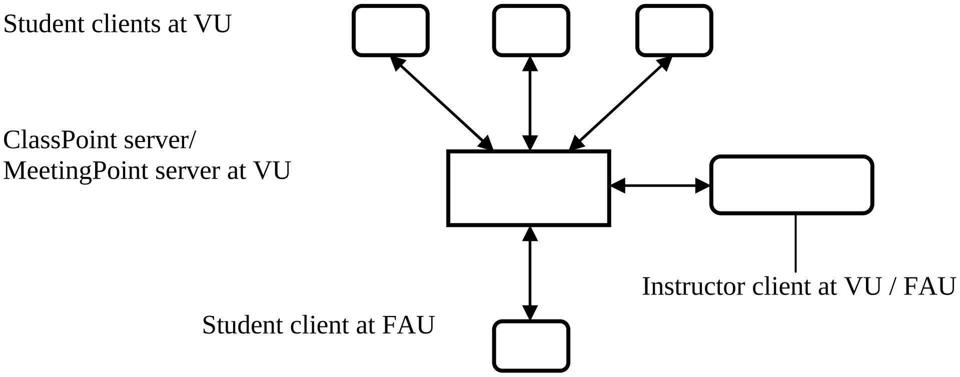


Figure 8. Experiments setup.

The ClassPoint server makes use of the MeetingPoint server. This server was setup at VU. Instructor and Student client software was installed at both FAU and VU.

Specifications of the LAN system, the client workstation, and the server are as follows.

<u>Client workstation used at FAU:</u>

- CPU: Pentium-II, 120 MHz.
- Memory: 64MB RAM.
- Operating System: Windows NT4.0 workstation.
- LAN connection: 10 Mbps Switched Ethernet.
- Internet connection to the LAN: T1 leased line.

<u>Client workstation used at VU:</u>

- CPU: Pentium Pro, 200 MHz.

- Memory: 32 MB RAM.
- Operating System: Windows95.
- LAN connection: 10 Mbps Switched Ethernet.

Server used at VU:

- CPU: Pentium Pro, 200 MHz.
- Memory: 64 MB RAM.
- Operating System: Windows NT4.0 server.
- LAN connection: 100 Mbps Switched Ethernet.
- Internet connection: 36 Mbps ATM Microwave link.

2.4 EXPERIMENTS AND QUALITATIVE RESULTS

The main aim of our experiments was to get a clear understanding of the capabilities of the ClassPoint software and its operation over the Intranet and the Internet environments. First, by running the Instructor client at FAU and Student clients at VU, some qualitative testing was performed. Nalin Sharda logged in as the instructor from FAU and conducted a distance education class for a group of students logged in from VU. We observed that simplex (one way) communication worked reasonably well. However, the turnaround time for half-duplex (two-way, one direction at a time) communication was too long, to consider the videoconference equivalent to a virtual face-to-face contact. For quantitative testing, Satish logged in as the Instructor client from VU, Krishna logged in as a Student client from FAU, and some other student clients logged in from VU. During these experiments quantitative measurements were made using the NetMedic tool installed on the clients at VU.

2.4.1 Experimentation over the Intranet

Experimentation over the Intranet involved installation of Instructor clients and the Student clients on user workstations, configuring them for optimum performance, and making appropriate measurements with the NetMedic tool while a videoconference session was in progress.

Audio and Video Conference: To observe the effect of additional students on the classroom operation in a simple videoconference, three sets of experiments were performed, with one, two and three students. Before starting the experiments ClassPoint Setup Assistant software was run on the clients to properly adjust the various settings like audio, video and microphone. Parameters such as Receive Rate, Send Rate, and CPU load were measured using NetMedic.

Whiteboard and Application-sharing: For this experiment, the instructor enabled the whiteboard and application-sharing feature of ClassPoint in his Instructor panel. Students were asked to join in collaborative data conferencing using the whiteboard. While the Instructor and Student clients were engaged in a conference the data rates and CPU load were noted using NetMedic. Next, the students were asked to share an application – such as, Microsoft Word or Power Point – with the instructor.

2.4.2 Experiments Performed on the Internet

Experiments performed on the Internet were almost similar to the experiments performed on the Intranet. However, the traffic conditions on the Internet were quite different. Network congestion and delays introduced due to the various components of the communication path, and geographical separation were some of the problems that affected the video quality on the Internet.

Audio and Video Conference: In the experiments performed on the Internet, the number of clients that can be introduced in the ClassPoint session were fewer as compared to those on the Intranet. Before establishing a videoconference between the Instructor and the Student clients the Setup assistant was run to fine-tune the audio and video devices. The size of the ClassPoint windows and the Student and Instructor panels were adjusted for good picture quality. While the videoconference session was in progress, data rates such as Receive Rate, Send Rate and the CPU load were measured using the NetMedic tool. In contrast to the experiments performed on the Intranet, the success rate of the experiments performed over the Internet was not good. While the Student client in Florida heard the Instructor client clearly, the Student client could not be heard properly in Melbourne. Audio was frequently breaking and the video streams were very slow. Sometimes the video froze completely. The Frame Rate was very low with little lip-synchronization. The experience on the Internet was a bit frustrating.

Whiteboard and Application Sharing: Experiments were performed with whiteboard and application sharing. The collaborative features worked satisfactorily. Some animation files were played at the Instructor client while the students were collaborating with the instructor. While audio and video communication over the Internet was of poor quality, collaborative tools worked well.

Distance Education Class: Nalin Sharda conducted a synchronous distance education class for his students at VU. Only one Student client was logged in from VU due to throughput limitations of the Internet. Nalin logged in as the Instructor client from FAU and Satish logged in as the Student client from VU. The computer screen of the Student client was projected on a large screen for the entire class to see. A half-hour lecture was conducted for the students of “Networked Multimedia” subject. The topic of the lecture was, “Distance Education Over the Internet”. Following the lecture, we had a question answer session. Students came to the Student client workstation, one at a time.

For the lecture, transmission was purely simplex. The quality of audio and video during the lecture was reasonably good; i.e. the students could understand the lecture content clearly. The question answer session used half-duplex communication. The turn-around time of the half-duplex communication was significant. This reduced the effectiveness of the virtual face-to-face contact. Figure 9 shows a couple of (digital) photographs taken while the actual videoconference session took place between Melbourne and Florida.

2.5 QUANTITATIVE RESULTS

In this section we present quantitative results obtained from the experiments described in the previous section.

Data, such as throughput measured at each client, can be useful in determining the network load and may be used for network capacity planning. This information can also help in setting appropriate limits on the maximum number of participants allowed on the server simultaneously. We used the bit rate measured at each client to calculate the burstiness of the data streams. Burstiness is the ratio of the Peak Bit Rate (PBR) to the Mean Bit Rate (MBR). This Information can be used to understand the data transmission requirements at particular nodes, and help in planning and designing networks for distance education.

Nalin Sharda and Satish Gangisetty

Sam Hsu and Satish Gangisetty

Figure 9. Videoconferencing between VU and FAU.

2.5.1 NetMedic.

NetMedic tool was used for measuring various parameters. This tool works well in the Windows and the Internet environments. It can be used to measure the Receive Rate, Send Rate, and CPU load (i.e. utilization). This tool can also be used to generate traffic reports, trace frequently visited sites, number of hops between the client and the server, slowest sites and also to keep track of the health of the computer. Although NetMedic provides a wide range of measurement techniques, only three of these were used in our experiments.

2.5.2 Data Rates and CPU Utilization.

Three experiments were performed on the Intranet, namely, Instructor client with one, two, and three Student clients. Figures 10 through 12 are the graphs of the data for the experiment with one Student client. These graphs depict the variations of Receive Rate, Send Rate and CPU utilization respectively, for the Instructor client.

Figure 10 shows how the Receive Rate (in Kbps) varied with time (in seconds). Over this period the Least Bit Rate (LBR) is 59 Kbps, Peak Bit Rate (PBR) is 108 Kbps, and Mean Bit Rate (MBR) is 71.8 Kbps. Burstiness of the Receive Rate is 1.5.

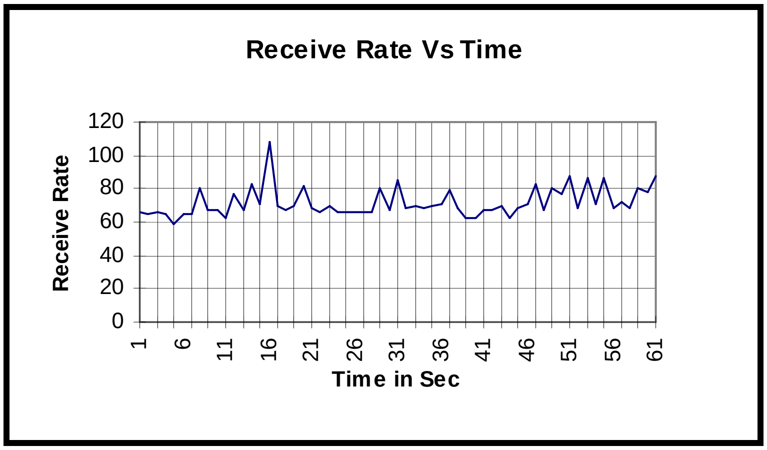


Figure 10. Receive Rate (Kbps) on the Instructor client connected to one Student client.

Figure 11 shows how the Send Rate varies with Time. LBR = 50 Kbps, PBR = 118 Kbps, MBR = 84 Kbps and burstiness = 1.4.

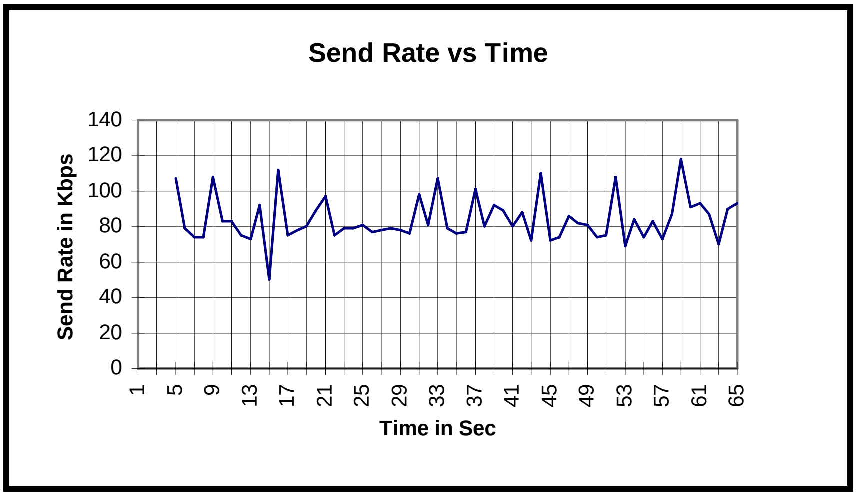


Figure 11. Send Rate on the Instructor client connected to one Student client.

Figure 12 shows CPU utilization variation with Time. It can be seen from the graph that CPU utilization varied between 30% and 79% with an average value of 48.5%, and burstiness of 1.63.

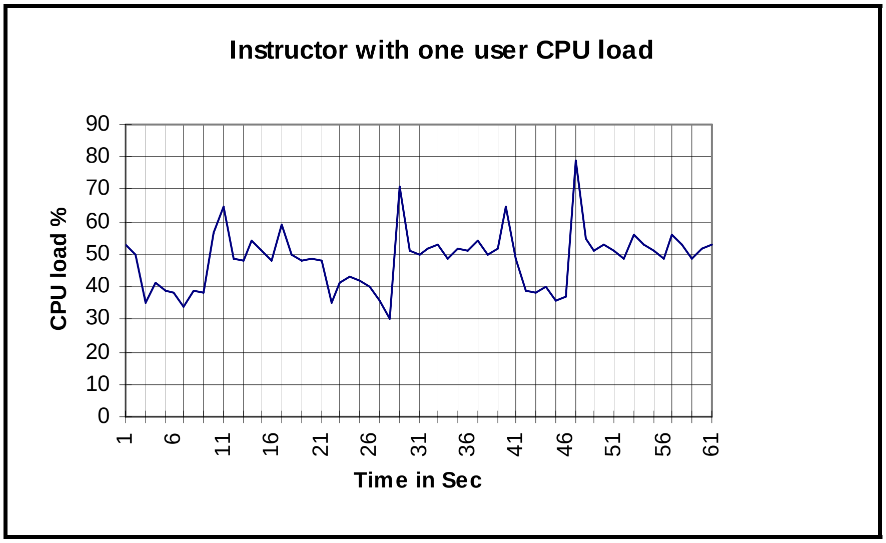


Figure 12. CPU Utilization on the Instructor client connected to one Student client.

Data Summary: Graphs of the data for two and three student experiments are not given, to avoid repetitious presentation. A summary of the results for the experiments performed on the Intranet is given in Table 1.

Table 1. Summary of Data Rates and CPU Utilization for Experiments over the Intranet

Parameter	1 Student	2 Students	3 Students
Receive Rate			
Least	59 Kbps	82 Kbps	89 Kbps
Peak	108 Kbps	122 Kbps	172 Kbps
Mean	71.8 Kbps	96.5 Kbps	126.8 Kbps
Burstiness	1.5	1.26	1.36
Send Rate			
Least	50 Kbps	70 Kbps	67 Kbps
Peak	118 Kbps	382 Kbps	114 Kbps
Mean	84 Kbps	87.8 Kbps	86.3 Kbps
Burstiness	1.4	4.35	1.32
CPU Load			
Least	30 %	39 %	36 %
Peak	79 %	83 %	90 %
Mean	48.5 %	59.8 %	60.4 %
Burstiness	1.63	1.39	1.4

Similar data for the experiments performed on the Internet are given in Table 2. In the Internet experiment only one Student client was logged on with one Instructor client. Two experiments (Ex-1, Ex-2) were performed. Data recorded for the Instructor and the Student clients and their average (Av.) values are given in Table 2.

Table 2. Summary of Data Rates and CPU Utilization for Experiments over the Internet

	Instructor Client			Student client		
Parameter	**Ex-1**	**Ex-2**	**Average**	**Ex-1**	**Ex-2**	**Av.**
Receive Rate	Kbps	Kbps	Kbps	Kbps	Kbps	Kbps
Least	81	87	**84**	85	84	**84.5**
Peak	126	134	**130**	122	120	**121**
Mean	101	97	**99**	98	97	**97.5**
Burstiness	1.24	1.38	**1.31**	1.24	1.23	**1.23**
Send Rate						
Least	1	68	**34.5**	33	69	**51**
Peak	103	112	**107**	92	99	**95**
Mean	86	80	**83**	72	78	**75**
Burstiness	1.2	1.39	**1.29**	1.27	1.27	**1.27**
CPU Load	%	%	**%**	%	%	**%**
Least	53	79	**66**	100	100	**100**
Peak	96	90	**93**	100	100	**100**
Mean	85	85	**85**	100	100	**100**
Burstiness	1.1	1.05	**1.07**	1	1	**1**

Data Analysis - Intranet Experiments: With three observation points, we can stipulate some conclusions. The main observations that we can make from the data presented in Table 1 are as follows.

All types of Receive Rates (least, peak, and mean) increase with the number of student clients. Though the increase is not proportional to the number of Student clients. Burstiness of the traffic, on the other hand, decreases with increasing number of Student clients. This implies that a high-speed connection will be better utilized with more students in a distance education class. In other words, the communication infrastructure cost can be amortized over a larger number of students lowering the per student cost.

The data for the Send Rate has similar characteristics, but for the anomalous jump in the traffic values for the two-student experiment. This could be due to higher level of movement in the picture frame.

Similarly, increase in the CPU utilization is apparent with increase in the number of students. Peak utilization for three-student experiment was 90%. This suggests that the CPU will get saturated if more students are added to the videoconference. In fact, we did observe that for four or more Student clients, the screen of the Instructor client froze every now and then. We attribute this to saturation of the CPU's processing power. Practical implication of this point is that the instructor must use a high powered CPU to have effective videoconference as the number of students increases.

Data Analysis - Internet Experiments: For experiments on the Internet we were restricted to only one Student client communicating with the Instructor client. With more than one Student client in the virtual classroom the system did not work properly. Therefore, we did not consider it worthwhile noting down traffic data. By comparing Tables 1 and 2 we can see that data rates obtained in the Internet experiment are of similar magnitude as those in the Intranet experiments. But the CPU utilization is much higher, especially for the Student client, which is 100%. A fully saturated CPU will lead to increased delay in processing and transmitting the data.

2.5.3 Transmission Delay

Good quality audio and video communications require low latency (accumulated delay) between the sender and the receiver. This is because audio and video loose relevance if they are not delivered in a timely manner. For a good quality audio and video the delay should not be more than 400 milliseconds (ms) [17]. The sender, the networking devices, and the receiver together introduce delays along the communication path [18].

Delay in audio and video transmission encountered over the Internet connection was measured. This delay dictates if the videoconference works well as virtual face-to-face contact, or not. The CPU load was also measured to identify situations where the bottleneck was the processing power of the CPU, and not the network.

Audio Delay Measurement: A simple experiment was performed in order to measure the delay in audio transmission. An agreed upon text trigger (TT) (e.g. Go) was typed in the chat window by the instructor. The student spoke a pre-decided audio trigger (AT) (e.g. “Hello”) on receiving the TT in his chat window. An electronic stopwatch was started at the very instant the TT was transmitted. When the instructor heard the AT, the stopwatch was stopped. The delay in transmitting the TT was considered negligible. Therefore the time on the stopwatch was noted as the delay in audio transmission. Delay was measured for three minutes over fifteen trails. Audio delay data is given in Figure 13.

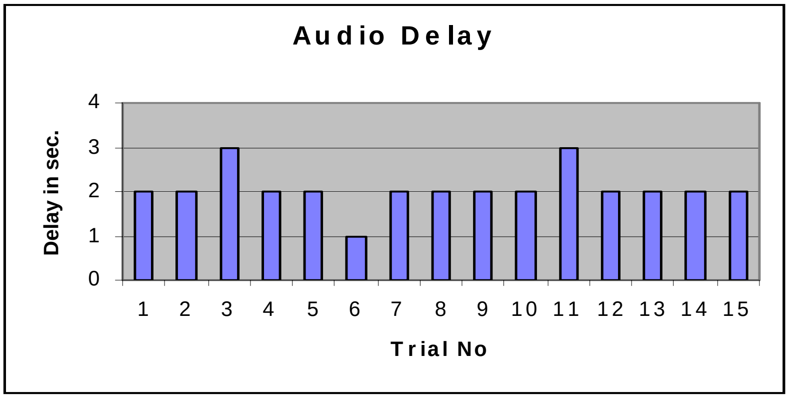


Figure 13. Transmission delay for audio.

Video Delay Measurement: Delay in video transmission was measured by a similar procedure. The instructor in Melbourne, transmitted a TT to the student in Florida. The student at Florida was asked to send a visual trigger (VT) (e.g. raise his forefinger) on receiving the TT. As the time delay in transmitting the TT was neglected, the time between sending the TT and receiving the VT was noted as the delay in video transmission. This delay was measured for four minutes, over twenty trials, as shown in Figure 14.

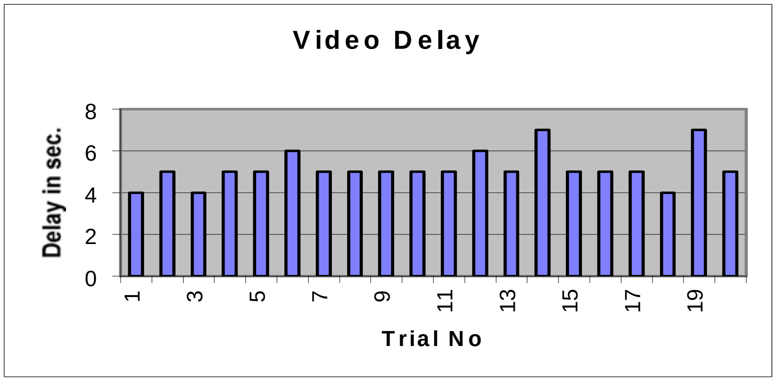


Figure 14. Transmission delay for video.

Delay Data Analysis: From Figure 13 we can see that the audio delay varied from 1 to 3 seconds. For most of the trials the audio delay was 2 sec. This is much greater than the acceptable value of 400 ms. We can see the variation in the video delay in Figure 14; it varied from 4 to 7 sec. For most of the trials the video delay was 5 sec. Not only is the video delay more than the acceptable value, but also more than twice the audio delay. This mismatch in audio and video delays led to severe lack of lip-synchronization.

One of the main reasons for these excessive delays is the lack of adequate bandwidth on many of the Internet connections. Moreover, the Internet is a public domain information network [18]. An end-to-end connection on the Internet passes through many intermediate links and nodes. Individual users do not have the ability to reserve service aspects, such as bandwidth and buffer space on these intermediate links and nodes. The TCP/IP suites of protocols used on the Internet were not designed to carry real-time audio and video data. Multimedia information networking systems [18] currently under development will alleviate some of these problems in the near future. Internet2 [19] is an effort aimed at developing an information superhighway for universities and research organizations. Such application specific information networks will also help in providing low delay connections over the Internet.

2.6 CONCLUSIONS

From our experiments we conclude that the videoconferencing tools such as ClassPoint provide excellent feature for synchronous distance education. It will be possible to use these tools for virtual classroom creation in the near future. But, at this point in time their suitability must be evaluated for specific environments. Operation of these tools is more successful on the Intranet environment; but that is not what distance education required. Operation of these tools over the Internet is far from satisfactory. Enhancements such as Internet2 will improve the situation. Further experiments are required to build a thorough understanding of the impediments in the deployment of videoconferencing for synchronous distance education.

3. VIRTUAL OFFICE HOURS USING NETMEETING

3.1 INTRODUCTION

The purpose of this project was to assess the feasibility of using off-the-shelf collaboration tools to implement the concept of virtual office hours.

The use of the Internet to enable communication between instructors and remote students in a format similar to that used for office hours is one of the features of many Web-based distance education systems. Virtual office hours (VOH) via the Internet can be established by using simple asynchronous tools such as e-mail or discussion lists. The interaction between instructor and students can be significantly improved by using synchronous, real-time tools. An easy and inexpensive way to enable such interactions can be obtained by using an off-the-shelf collaboration software.

We have assessed the usefulness and suitability of a specific software package, Microsoft® NetMeeting™, for the implementation of virtual office hours. The product's main features, some of its pros and cons, the experiments performed to evaluate its use for virtual office hours and corresponding results and lessons learned are described next.

3.2 WHY VIRTUAL OFFICE HOURS

The use of the Web for distance education has grown exponentially in the past few years. An ever-growing number of Web-based courses in virtually any field of knowledge are now available worldwide. Moreover, conventional, live courses are also incorporating Internet-based features, such as compilation of relevant course information into HTML pages, electronic submission of assignments, and e-mail communication between students and instructors.

One of the most important features of any distance education system is the ability to allow remote students to reach their instructors and interact with them in a format similar to the well-known concept of office hours. The simplest way of implementing the VOH concept is to enable e-mail communication between students and instructor. An alternative option consists of offering the students the possibility of formulating their questions using a Web-based form, such as in [20]. A natural extension of the concept is the use of discussion lists, where students can benefit from other students' questions and previous discussions on the subject. All these methods of communication between students and instructors are asynchronous by nature, and their effectiveness will depend strongly on how promptly the questions get answered.

The VOH concept was made popular by the Department of Chemistry and Biochemistry at UCLA [21] and was later extended to many other colleges and universities, such as the Santa Monica College [22]. Their VOH project has got a very high approval rate from students [21][23] who appreciate the availability of course material in electronic format as well as the possibility of interacting with instructors out of class or office hours. At a larger scale, students in many other colleges and universities are having access to similar systems and their response has been consistently positive [24].

We believe that virtual office hours should go beyond asynchronous interaction and decided to test possible tools and techniques to implement real-time, synchronous interaction between students and instructors using a commercial (off-the-shelf) collaboration tool.

3.3 WHY NETMEETING

There are many conferencing tools available, such as White Pine's CU-SeeMe, PictureTel's LiveShare Plus, or Microsoft NetMeeting. Some of them, e.g. White Pine's ClassPoint, target specifically at the distance education market. These tools usually provide highly interactive, synchronous, bi- or multidirectional features, e.g. audio and video conferencing, and multipoint data conferencing using tools such as chat rooms, shared clipboard, whiteboard, and file transfer capabilities. We decided to use Microsoft NetMeeting for a number of reasons described later in this chapter. At the same time, we carried out experiments using White Pine's ClassPoint (see Section 2 of this Chapter) and continued developing our own Web-based collaboration tools [1] [25].

Microsoft NetMeeting can be briefly described as a collaboration tool that combines voice and data communications, video, real-time application sharing, file transfer, a full-featured shared whiteboard, and text-based chat [26]. NetMeeting is targeted at home users, as well as small and large organizations and claims to allow users to “take full advantage of the global reach of the Internet or corporate intranet for real-time communications and collaboration.”[27] Connecting to other NetMeeting users is made easy with the Microsoft Internet Locator Server (ILS), enabling participants to call each other from a dynamic directory within NetMeeting or from a Web page. Connections can also be established by calling the other party’s IP address. While connected on the Internet or corporate intranet, participants can communicate with audio and video, work together on virtually any 32-bit Windows-based application, exchange or mark-up graphics on an electronic whiteboard, transfer files, or use the text-based chat program.

Some of the most prominent features of NetMeeting 2.1 are [27]:

- **Internet phone/H.323 standards-based audio support**: real-time, point-to-point audio conferencing over the Internet or corporate Intranet. Half-duplex and full-duplex audio support for real-time conversations, automatic microphone sensitivity level setting, and microphone muting, which lets users control the audio signal sent during a call. This audio conferencing supports network TCP/IP connections. Support for the H.323 protocol [28][29] enables interoperability between NetMeeting and other H.323-compatible audio clients.

- **H.323 standards-based video conferencing** including the following features:

  (a) Participants can switch audio and video to another person during a meeting. This feature compensates for the impossibility of multipoint audio and video conferencing.
  (b) Users can dynamically adjust the video image quality and change the size of the video window, balancing the need for higher quality or faster performance.
  (c) In the NetMeeting main window, the video preview and receive windows are integrated on the Current Call window. Users can view these video windows from Current Call, or drag them to a different location on the desktop.
  (d) Users can choose whether or not to transmit video immediately when a call starts. Also, they can pause or resume sending or receiving video by pressing a button in the video window frame.
  (e) NetMeeting automatically balances the performance and quality of video during a meeting based on the speed of the network connection, providing the highest quality, lowest bandwidth video capabilities.
  (f) Administrators can control access to video features using NetMeeting custom settings.
  (g) On MMX-enabled computers, NetMeeting uses the MMX-enabled video codecs to improved performance for video compression and decompression algorithms.
  (h) Support for H.323 conference servers and gateways (currently being developed by leading vendors) will enable NetMeeting users to participate in meetings with multiple audio and video connections.
  (i) H.323 standard-based video technology is also compliant with the H.261 [30] and H.263 [31] video codecs.

- **Intelligent Audio/Video Stream Control**, which automatically balances the load for network bandwidth, CPU use, and memory use. This intelligent stream control ensures that audio, video, and data are prioritized properly, so that NetMeeting maintains high-quality audio while transmitting and receiving data and video during a call.

- **Multipoint data conferencing**: Two or more users can communicate and collaborate as a group in real time. Participants can share applications, exchange information through a shared clipboard, transfer files, collaborate on a shared whiteboard, and use a text-based chat feature. Also, support for the T.120 data conferencing standard enables interoperability with other T.120-based products and services. The multipoint data conferencing features are listed in Table 3.

NetMeeting provides a broad support of international standards, to ensure interoperability among solutions from different vendors.

3.3.1 NetMeeting pros and cons

The following *pros* and *cons* are compiled based on data from comparative reviews, threaded discussions, relevant Web pages and the authors' experience using NetMeeting in their classes.

Pros

- NetMeeting users can talk, send video, type text and share applications all at the same time [26].
- The main tool as well as its Software Development Kit (SDK) and Resource Kit (for Network Administrators) are free and can be downloaded from Microsoft Web site [32].
- NetMeeting is continuously being improved and its basic architecture is being widely supported.
- NetMeeting can be expanded / integrated with other applications using its SDK.

- Broad support of international standards for easy interoperability with solutions from different vendors.
- NetMeeting provides a good degree of privacy during a call (using the “do not disturb” option).
- NetMeeting supports application sharing in multiparty calls. Any collaborating participant can assume mouse control and work in the shared application while others watch. Application sharing has proven to be NetMeeting’s “killer feature” [26].

Table 3. Features of Multipoint Data Conferencing.

Application sharing	A user can share a program running on one computer with other participants in the conference. Participants can review the same data or information, and see the actions as the person sharing the application works on the program.
Shared clipboard	The shared clipboard enables a user to exchange its contents with other participants in a conference using familiar cut, copy, and paste operations.
File transfer	Users can send a file in the background to one or all of the conference participants. When one user drags a file into the main window, the file is automatically sent to each person in the conference, who can then accept or decline receipt. This file transfer capability is fully compliant with the T.127 standard.
Whiteboard	Multiple users can simultaneously collaborate using the whiteboard to review, create, and update graphic information. The whiteboard is object-oriented (as opposed to pixel-oriented), enabling participants to manipulate the contents by clicking and dragging with the mouse. In addition, they can use a remote pointer or highlighting tool to point out specific contents or sections of shared pages.
Chat	A user can type text messages to share common ideas or topics with other conference participants, or record meeting notes and action items as part of a collaborative process. Also, participants in a conference can use chat to communicate in the absence of audio support.

Cons

- NetMeeting’s audio and video features work only between two participants, so a three-way call cannot be established.
- NetMeeting is very CPU intensive and requires a powerful machine, especially for audio and video conferencing. The recommended hardware includes a Pentium-MMX 200 MHz or better, at least 32 MB of RAM, 40 MB of free disk space, and good quality peripherals (sound card, video card, microphone or headset, camera, and modem) [26].
- Audio pauses, delays, and momentary loss of full-duplex capabilities are frequent. Moreover, delays, variable latency (jitter) and uneven packet transmission rates (common to any IP-based transmission) may cause deterioration of audio, video, and whiteboard actions.
- Video frame size is too small and frame rate is too low to make video communication a fully enjoyable experience.
- Problems using NetMeeting in intranets protected by firewalls have been reported [33]. Directions of how to get around with this problem can be found at [34] and in chapter 11 of [26].
- NetMeeting has shown security leaks, which can pose great risk to users’ data and privacy. An example of a problem in this category is the so-called "Speed Dial" issue [35].

- NetMeeting 2.1 contains a useful whiteboard module that does not comply with T.126, the object-exchange standard under T.120. However, Databeam has produced a tool called MeetingTools [36], which installs on top of NetMeeting and enhances NetMeeting calls. With MeetingTools, NetMeeting users can drive "massive-point" conferences, where hundreds of simultaneous users view the activity from their Web browsers.
- NetMeeting setup is time-consuming and non-technical users can easily get frustrated.
- NetMeeting can be seen as part of the Microsoft monopolizing strategy, and its seamless integration with other Microsoft products (particularly Internet Explorer, Outlook, Windows Explorer, and Exchange) is controversial. Along the same reasoning, one may wonder how long will the product still be offered free of charge.

3.4 EXPERIMENTS

NetMeeting has been used by the authors to implement the concept of virtual office hours. Our aim is to extend to distant students via the network the one-to-one interaction of typical office hours meetings. To evaluate the usefulness of specific NetMeeting features for the purpose of distance education, particularly for virtual office hours, students from two sections of the “Introduction to Microprocessor Systems (CDA3331)” course, offered by one of the authors during the Spring’99 semester have been selected to participate in evaluating those aspects.

It is expected that the level of interaction (e.g., text only, one- or two-way audio/video) be limited mainly by the students’ hardware and networking capabilities. However, to what extent the impact that different hardware, software, and especially the quality of the networking connection would have on the overall effectiveness of the approach has remained as an open question that has motivated the authors to design and carry out actual experiments. So far, eight separate experiments in rather different environments, ranging from low-speed modem dial-up connection from students’ homes to LAN-based connection with two-way audio and video capabilities between one of our computing laboratories and the instructor’s office have been conducted. All experiments have been carried out on a one-to-one and student-to-instructor basis.

In each experiment, a student initiated a dialog by calling the instructor’s office using NetMeeting (dialing the instructor’s network IP address) in a way similar to a phone call. After connection was made, efforts would be directed at attempting to use all possible features of NetMeeting initially. Whenever a specific feature would exhibit problems (e.g. two-way audio), we would fall back to the minimal interaction, i.e. text-based chat, and agree on using whatever tools were available to achieve a satisfactory two-way communication. The subject of the conversations was always related to the course, simulating a live office hour meeting, lasting between 10 to 30 minutes.

3.5 RESULTS

The results detailed below refer to a universe of eight experiments, each involving one student and one instructor in a one-to-one and real time interaction. A total of eight different students participated in the experiments. Since this number may not represent a significant sample space of the total student population, we emphasize their importance in a qualitative, rather than quantitative way.

- **Connection**: we managed to achieve a good mix of different connection conditions: half the students used the department LAN, calling the instructor’s office from one of our labs. The other half was equally spread between modem, cable modem, and T1 (enterprise) Internet connections.

- **NetMeeting's features**: students evaluated eight features for their quality and usefulness for the VOH concept using a 0-3 scale (0 = very bad; 3 = very good). The distribution of grades for each feature is summarized in Figure 15. From that diagram several conclusions can be drawn:
  (i) The most consistent feature (which was always available, regardless of technical difficulties) was the text-based chat. All students used it and graded it "very good".
  (ii) The shared whiteboard (see Figure 16 for an example of whiteboard contents saved into a file for possible reuse) got a very positive response from all students who used it.
  (iii) The file transfer feature got positive evaluations from the few students who used it.
  (iv) The application-sharing feature was highly appreciated by the few students who tried it.
  (v) Audio and video capabilities got all possible grades, showing a high dependence on the quality of the equipment and the network connection.

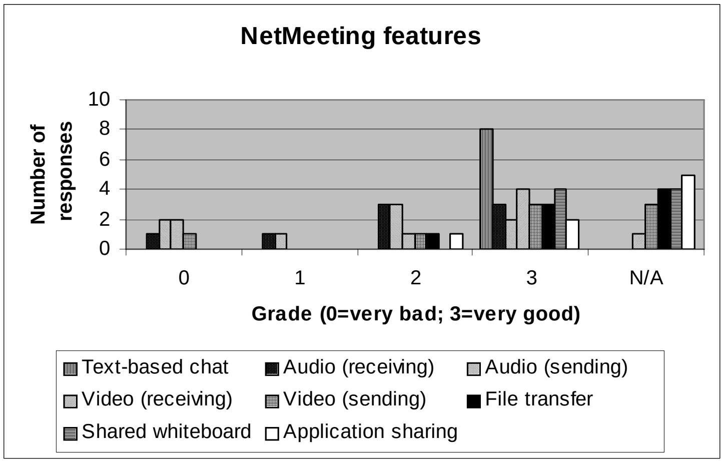


Figure 15. Summary of evaluation of NetMeeting's features.

- **General evaluation**: 75% of the students considered NetMeeting a suitable tool to implement the VOH concept. All but one rated the tool as "very useful" to help in distance learning, in general. Overall, three quarters of the students considered the VOH concept (very) useful.

3.6. LESSONS LEARNED, CONCLUSIONS, AND RECOMMENDATIONS

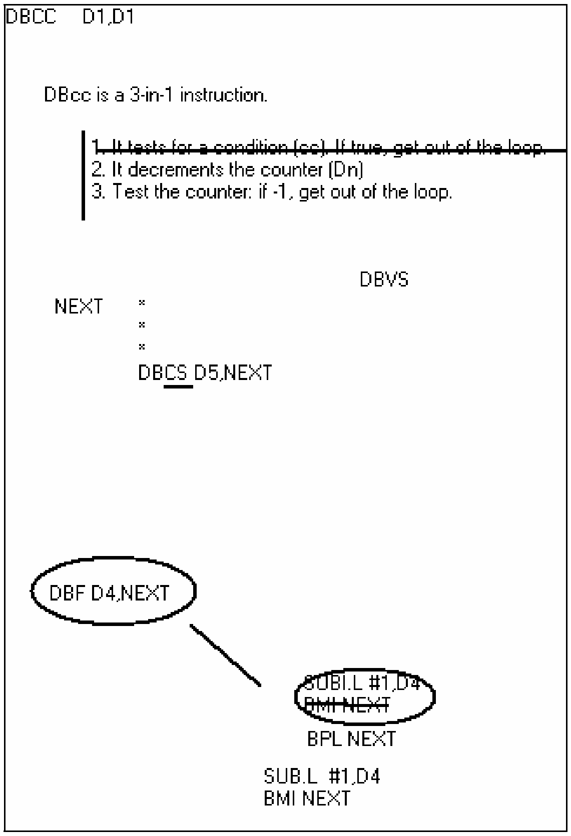
The VOH experiments have been evaluated to be satisfactory both from the instructor's as well as the students' perspectives. Some of the lessons we have learned are:

- NetMeeting setup can be time-consuming and many students got discouraged after a few unsuccessful attempts.
- Students (particularly Computer Science and Engineering students) are always supportive to the introduction of new technology into the educational process and eager to use it to its maximum possible extent. One clear example is the preference for NetMeeting calls over telephone calls to the instructor's office, despite all the technical difficulties with two-way audio connection over the Internet.
- There is a clear and strong dependence between the quality of the connection and the overall satisfaction with the tool. Students who tried it from one of the labs during Spring Break (network traffic was low those days in contrast to car traffic by the beach) had a

completely different experience than those who attempted to use NetMeeting from home, in the middle of a busy working day.

- For the purposes of VOH, NetMeeting features can be divided into essential, extremely useful, and accessory. In the first group we'd include text-based chat and two-way audio. In the second group, shared whiteboard and application sharing. File transfer and video communication would belong to the last group.

We believe the overall results of our first experiments were very positive and encouraging. They have confirmed our expectations as to whether NetMeeting could be used to implement the VOH concept. We plan to extend the use of NetMeeting for virtual office hours during the upcoming semester and will keep an eye on possible third-party products that can make the experience even more enjoyable.



DBCC D1,D1

DBcc is a 3-in-1 instruction.

~~1. It tests for a condition (cc). If true, get out of the loop.~~
2. It decrements the counter (Dn)
3. Test the counter: if -1, get out of the loop.

DBVS

NEXT *
*
*
DBCS D5,NEXT

DBF D4,NEXT

SUBI.L #1,D4
~~BMI NEXT~~
BPL NEXT

SUB.L #1,D4
BMI NEXT

Figure 16. Whiteboard contents.

References

[1] S. Hsu, A. Pandya, and R. Patel, "Yet Another Virtual Classroom Over the Internet," *Proc. of 1998 ASEE/IEEE Frontiers in Education Conference (FIE '98),* November 1998, Tempe, Arizona, pp. 717-721.

[2] N. Sharda, S. Gangisetty, K. Adusumilli, and S. Hsu, "Application of Video Conferencing to Distance Education: A Case Study with the ClassPoint Software," *Proc. of SCI'99, World Multi-conference on Systems, Cybernetics and Informatics,* August 1999, Orlando, Florida.

[3] Oge Marques, Sam Hsu and Nalin Sharda, "Virtual Office Hours Using NetMeeting," *Proc. of SCI'99, World Multi-conference on Systems, Cybernetics and Informatics*, August 1-5, 1999, Orlando, Florida.

[4] Yahoo!-Education:Distance Learning: Colleges and Universities, http://www.yahoo.com/Education/Distance_learning /Colleges_and_Universities/.

[5] ICDL ONLINE, http://www-icdl.open.ac.uk/ index.htm.

[6] M. Turoff, "Designing a Virtual Classroom," *Proc. of 1995 International Conference on Computer Assisted Instruction ICCAI '95*, March 1995, Hsinchu, Taiwan. http://www.njit.edu/CCCC/VC/Papers/ Design.html.

[7] D. Dwyer, K. Barbieri, and H.M. Doerr, "Creating a Virtual Classroom for Interactive Education on the Web," *Proc. of 3rd International WWW Conference*, Darmstadt, Germany, Apr. 10-14, 1995, http://www.tc.cornell.edu/~dwyer/ Overheads/Virtual.Classroom/ctc.virtual.class.html.

[8] T.D.C. Little and D. Venkatesh, "The Use of Multimedia Technology in Distance Learning," *Proc. IEEE Intl. Conf. on Multimedia and Networking*, September 1995, pp. 3-17.

[9] R.T. Kouzes, J.D. Myers, and W.A. Wulf, "Collaboratories: Doing Science on the Internet," *IEEE Computer*, Aug. 1996, pp. 40-46.

[10] M-C Lai, B-H Chen, and S-M Yuan, "Toward a New Educational Environment," *Proc. 4th International WWW Conference*, Boston, Massachusetts, December 1995. http://www .w3j. com/1/lai.238/paper/238.html.

[11] K. Maly, H. Abdel-Wahab, C. Overstreet, J. Wild, A. Gupta, A. Youssef, E. Stoica, and E. Al-Shaer, "Interactive Distance Learning over Intranets," *IEEE Internet Computing*, Vol. 1, No. 2, Jan/Feb 1997, pp. 60-71.

[12] R. Patel, "VCOIN: Virtual Classroom Over the Internet," Master's thesis, Florida Atlantic University, April 1998.

[13] http://www.wpine.com/Products/ClassPoint/index.html.

[14] G. Gagliardi, "*Client / Server Computing*," New Jersey: Prentice Hall, 1994.

[15] http://www.wpine.com/Products/MeetingPoint/index.html

[16] B. O. Szuprowicz, "*Multimedia Networking*," McGraw-Hill, 1995.

[17] F. Fluckiger, "*Understanding Networked Multimedia: Applications and Technology"* New Jersey: Prentice Hall, 1995.

[18] N. K. Sharda, "*Multimedia Information Networking*," New Jersey: Prentice-Hall, 1999.

[19] http://www.internet2.edu.

[20] Virtual Office Hours Page http://www.isop.ucla.edu/ids/mailform/mailform.htm.

[21] Virtual Office Hours, White Page http://xenon.chem.ucla.edu /uclavoh/docs/white-page.html.

[22] SMC Virtual Office Hours, http://voh.smc.edu/.

[23] J. M. Chen, "Virtual office hours connect professors, students" http://www.dailybruin.ucla.edu/DB/Issues/96/2.23/news.virtual.html

[24] G. Landgraf, "Virtual office hours," http://www.student.com/article/emaileducation

[25] S. Hsu and K. K. Adusumilli, "Softboard - An Online Blackboard", *Proc. AACE/WebNet '98*, November 1998, Orlando, Florida, Session 829.

[26] B. Summers, "*Official Microsoft NetMeeting Book*," Redmond, WA: Microsoft Press, 1998.

[27] NetMeeting Resource Kit, http://www.microsoft.com/netmeeting/reskit/.

[28] Draft ITU-T Recommendation H.323V2 (1997): "Line Transmission of Non-Telephone Signals."

[29] A Primer on the H.323 Series Standard http://www.databeam.com/h323/h323 primer.html

[30] ITU-T Recommendation H.261 (1993): "Video Codec for audiovisual services at pX64 kbit/s."

[31] ITU-T Recommendation H.263 (1995): “Video Codec for narrow telecommunications channels at < 64 kbit/s.”

[32] NetMeeting Home, http://www.microsoft.com/netmeeting.

[33] WWWDEV Archived Mailing List: Net-Meeting - Discussion thread starting at: http://leahi.kcc.hawaii.edu/org/wwwdev/logs/0811.html.

[34] How to Establish NetMeeting Connections Through a Firewall**,** http://support.microsoft.com/support/kb/articles/q158/6/23.asp.

[35] NetMeeting Speed Dial issue, April 21, 1998 **,**http://www.microsoft.com/windows/ie/security/ netmbuff.asp.

[36] DataBeam MeetingTools, http://www.databeam.com/meetingtools/.