International Conference on Engineering Education and Research "Progress Through Partnership"
© 2004 VSB-TUO, Ostrava, ISSN 1562-3580

Evaluating the Quality of Service of a Floor-Controlled Centralized Conference System for Distance Learning

George Marcel M. A. SMETANA; Reinaldo MATUSHIMA; Regina Melo SILVEIRA; Tereza Cristina M. B. CARVALHO; Wilson Vicente RUGGIERO
Laboratório de Arquitetura e Redes de Computadores (LARC) – Escola Politécnica (EP) – Universidade de São Paulo (USP), Av. Professor Luciano Gualberto, travessa 3, n. 158, sala C1-46 – CEP 05508–900 – São Paulo – SP – Brazil, {gsmetana,matushim,regina,carvalho,wilson}@larc.usp.br, <URL: http://www.larc.usp.br>

KEYWORDS: *Distance learning, floor control, videoconference, quality of service*

ABSTRACT: *This paper describes a floor-controlled centralized conference system being developed for LARC (Laboratory of Computer Architecture and Network – Escola Politécnica – Engineering School) distance learning tool*

Floor control was used to regulate interaction and access to the audio channel and to restrict network bandwidth usage. A floor control protocol and a light session protocol were developed. Implementation was made using web technology in order to ease integration with web-based systems. This system runs over IP unicast communication infrastructure. Some experiments were made in order to evaluate the quality of service of a centralized conference system for distance learning. The quality of service parameters evaluated for the conference system user were usability, interactivity, voice subjective quality and bandwidth needed, while the parameters evaluated for the conference system owner were bandwidth, processing costs and system scalability. This paper aims to present the results of such experiments. In the first one, floor control was tested as a mechanism for restricting bandwidth consumption in centralized conferences and the bandwidth needed by the user, by the multipoint control unit (MCU) and by the conference system owner were evaluated. In the second one, delay and packet loss were seen through the use of a real world scenario (remote tests), impacting the voice subjective quality and the interactivity among conference participants. In the last one, a subjective experience reveals information about the usability of the system. These results have taken as basis to improve the current conference system and the modus operandi of the distance learning system as a whole.

1 INTRODUCTION

As distance learning usage grows, some of their tools try to incorporate virtual collaborative environments technologies and techniques to overcome natural barriers imposed by spatial separation, like lack of human-to-human interaction. Nonverbal communications, as gestures and eye-gaze, play important roles in face-to-face interactions. Videoconferencing helps raising interaction, but network delay and poor video quality harden attaining the same effect as physical presence and mutual awareness.

A virtual classroom for distance learning is one of the paradigms for collaborative environments proposed in [Qiu, Kuhns and Cox (2002)]. In this paradigm, the interaction among participants is asymmetric, that is, the audio and video streams of the teacher are transmitted to the students and, occasionally, a student is allowed by the teacher to address the class. The virtual classroom paradigm is a particular specialization of the chaired conference paradigm, the teacher behaving like the chairman and the students, like the audience.

This paper describes a floor controlled centralized conference system developed for LARC's distance learning tool and its quality of service evaluation, regarding the user and conference system owner aspects. Floor control was used to regulate interaction and access to the audio channel and to restrict network bandwidth usage. A floor control protocol and a light session protocol were developed. Implementation was made using web technology in order to ease integration with web-based systems.

Some experiments were made in order to evaluate the quality of service of the centralized conference system for distance learning. The quality of service parameters evaluated for the conference system user were usability, interactivity, voice subjective quality and bandwidth needed, while the parameters evaluated for the conference system owner were bandwidth, processing costs and system scalability.

In Section 2, background information and some relevant related works are presented. Section 3 describes the system architecture, environment and implementation. Section 4 brings some experimental results regarding the floor control mechanism, the network bandwidth consumed and user experience and Section 5 discloses conclusions and future work.

2 BACKGROUND

Information about the Real-Time Transport Protocol (RTP) can be found in [Schulzrinne et al. (1996)]. Floor control can be found in [Dommel and Garcia-Luna-Aceves (1997)] and [Crowley, Milazzo, Baker, Forsdick and Tomlinson (1990)]. Network address translators (NATs), network address and ports translators (NAPTs), their operation and problems associated can be found in [Holdrege and Srisuresh (1999, 2001)] and [Senie (2002)].

Although IP multicast consumes less bandwidth then IP unicast in multipoint conferences, it is not always possible to use it, because both the network infrastructure and the multimedia conference system (hardware and software) must support multicast traffic. Further, global scope conferences must have global multicast addresses, which are difficult to be obtained. IP unicast does not have these drawbacks, but consumes more bandwidth, a problem that can be mitigated by the use of floor control protocols, limiting the number of simultaneous senders.

Controlling participants' audio and video transmissions is necessary to limit the bandwidth consumed in videoconferences, but may reduce interactivity among the participants. For instance, in a centralized audio conference with N participants without floor control, the multipoint control unit (MCU) would receive N audio channels and forward each of them to $N - 1$ participants, resulting in $N (N - 1)$ audio channels being sent. Each participant would receive $N - 1$ audio channels. Without floor control, bandwidth rapidly becomes the main bottleneck for the scalability of conferences. Voice activity detection (VAD) and mixing help but do not eliminate the need of floor control for reducing the consumed bandwidth.

Quality of service parameters can be divided into system user and system owner.

For the system user, four characteristics of the conference system are important: usability, interactivity, voice subjective quality and bandwidth needed. Usability depends on how well the human-machine interface and the whole system were designed, while interactivity depends on the system's response time, in this case, network delay. Voice subjective quality is affected by the codec used, by packet loss and by delay. For the user, the bandwidth needed is important in order to know if he/she is able to use the system.

For the system owner, three characteristics are important: bandwidth, processing costs and system scalability, which depend on bandwidth usage/availability and processing power. Bandwidth is still a costly resource, therefore, as lower its usage, the better for the company.

3 SYSTEM ARCHITECTURE, ENVIRONMENT AND IMPLEMENTATION

The conference system was built in order to reproduce the conversational dynamics of a classroom, where the interaction among participants is asymmetric, that is, the audio stream of the teacher is transmitted to the students and, occasionally, a student is allowed by the teacher to address the class, in order to ask/answer a question or to make comments. The use of the conference is not restricted to a virtual classroom, but it can be used in any chaired conference, where one participant has the social role of the moderator, regulating the interaction among the other participants.

The conference system can be used over local and wide area networks based on the TCP/IP architecture. In real world situations, remote users may be behind firewalls, NATs or NAPTs, hardening communication. The conference system requires firewalls to be configured to allow inbound and outbound UDP traffic.

The system architecture is shown in Figure 1 and its components are: web server, database server, MCUs server, MCUs, user web page and a user Java applet using the Java Media Framework (JMF).

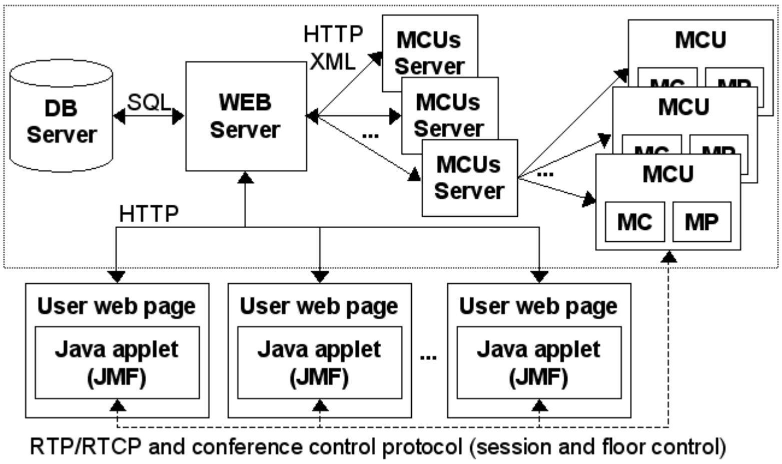

Figure 1 – System architecture.

The web server hosts the management website and the conference website. User authentication is accomplished by the website design, with passwords and logins. Access to the management website is restricted to the system administrator. At the management website, the system administrator is able to: create, modify, monitor, delete, start and finish virtual conference rooms; create, modify and delete users; associate users to virtual conference rooms.

The database server records all these information.

Each virtual conference room has its own MCU and a set of participants. When a virtual conference room is started or finished, the management website sends a HTTP request to the MCUs server with the corresponding action, asking it to start or finish the associated MCU. The MCUs server sends a XML response indicating if the action requested was completed successfully or not. The MCUs server instantiates and destroys MCUs objects.

The MCU establishes point-to-point communication with each participant. The multipoint controller (MC) handles and forwards session and floor control packets, while the multipoint processor (MP) deals with the RTP/RTCP audio packets. The MCU is completely codec independent, since it does not realize mixing, or any codec conversion, which are processor intensive operations. A codec independent MCU is able to scale better than a dependent one.

The conference website displays the available virtual conference rooms to the user. The virtual conference room is represented by the user web page, which provides the human-machine interface.

The Java applet inside the user webpage provides the multimedia functionality and network communication with the MCU. The Java applet implements the RTP/RTCP, the session and the floor control protocols. Audio and text communications are available, but audio is restricted to the floor holder.

3.1 System Requirements and Performance Issues

Table 1 shows server requirements, while Table 2 shows user requirements.

Table 1. Server requirements.

Software requirements	Microsoft Windows 2000 Server Web server: Microsoft Internet Information Server 5.0 Database server: Microsoft SQL Server 2000 Sun Microsystems Java Virtual Machine (JVM) 1.4.2 or superior
Hardware requirements	PC with 600 MHz or higher processor clock speed 512 MB of RAM or higher 100 MB of available hard disk space (plus disk space requirements for each software) Network adapter
Network requirements	Upstream: at least 50 kbps per user in conference Downstream: at least 100 kbps per conference

Table 2. User requirements.

Software requirements	Windows 98, 2000 or XP Internet Explorer 5.5 or superior Sun Microsystems Java Virtual Machine (JVM) 1.4.2 or superior Sun Microsystems Java Media Framework Windows Performance Pack (JMF) 2.1.1e or superior
Hardware requirements	PC with 350 MHz or higher processor clock speed 128 MB of RAM or higher 100 MB of available hard disk space Sound card: PCI, full-duplex and Creative Labs Sound Blaster compatible Video card: PCI or AGP, with 4 MB or greater 56 kbps analog modem, xDSL, Cable Modem or network adapter Microphone and headphone (speakers are not recommended)
Network requirements	Internet access with at least 50 kbps available to the uplink and downlink

Server-side scripts were implemented using Microsoft Active Server Pages. Client-side scripts were implemented using JavaScript.

The MCU and the MCUs server were implemented in Java and thus can be hosted by several platforms.

Available memory is crucial for the user Java applet, since it requires real-time processing. When the Java applet demands the use of virtual memory, the performance drops causing audio packets to be lost.

Network bandwidth is critical to the conference system. If there are N participants in a conference, the MCU must forward the audio and conference control packets received to $N - 1$ participants. For this reason, the server hosting the MCUs server must have a high-speed Internet connection. If the conference system is supposed to serve up to five simultaneous conferences with ten participants each, there would be five audio channels being received, forty-five audio channels being transmitted and fifty control channels being received and transmitted simultaneously by the MCUs.

Because JMF does not support traffic prioritization (the *Type of Service* – TOS or *Differentiated Services* – DiffServ fields in the IPv4 header), the current version of the conference system does not implement it. Attaining quality of service is difficult without traffic prioritization and when IP addresses and UDP/TCP ports are unknown.

3.2 Conference Control: Floor and Session Control

Floor and session control were built as a single conference control protocol transported over UDP. A control type identifier in the packet header differentiates floor and session control functionality. Each packet has a user identifier. Users can play the role of ordinary participants or room moderator (chairman). Floor and session control functionalities are available through and represented by the user's human-machine interface.

The conference control protocol permits ordinary participants to: ask to enter a conference room; leave the conference room; request the floor; give up the floor requisition and return the floor, after it has been granted to him/her.

Moderators control session admission and floor access and are able to: start, finish and configure conference rooms; admit, reject and expel participants; grant or deny floor to any participant; take back the floor at anytime; request the floor, just to tell the current floor holder that he/she wants it back; permit or deny the MCU to forward the audio channel to any participant and elect a new moderator.

The conference room is represented in the user's interface by pictures of each participant in the conference. When the user clicks on someone's picture, a menu presenting available actions and information pops up. The available actions vary according to the user role in the conference and its current indication. Coloured frames around user's picture represent user indications. Audio cues are played to the user, whenever the floor is granted to or taken from him/her or when the user is requesting admission in the conference. Audio cues are important, mostly when user attention is needed in different parts of the screen or in different windows.

Valid floor and session control indications and their associated audiovisual cues in the user interface are: *normal* (the participant is in the conference and there is no other indication more appropriated. No frame. If the previous user indication was *holdingFloor*, he/she hears a "punch"); *requestingFloor* (the participant is requesting the floor. Green frame); *holdingFloor* (the participant holds the floor. Red frame. The user hears NBA's pipe organ, the "ascend to charge" sound); *noInformation* (the link between the participant and the MCU may be down. Yellow frame); *noAudio* (the moderator has ordered the MCU not to forward audio packets to the participant. Cyan frame); *expelled* (the participant was expelled from the conference and can no longer request admission until the moderator does not let him to come back. Black frame and picture with transparency); *admissionPending* (the participant is requesting admission in the conference. Blue frame. The participant hears the phone ringing out, while the moderator hears the phone ringing in) and *absent* (he participant has a reserved chair in the conference room, but has not entered the conference yet. No frame and picture with transparency).

Figure 2 shows a screenshot of the user web page representing a virtual conference room with ten participants and with text communication.

Figure 2 – Screenshot of the user web page representing a virtual conference room (human-machine interface in Brazilian Portuguese).

Any user action implies the MCU to react accordingly. For example, if the moderator grants the floor to a participant, the MCU forwards the granting to all participants, so they can keep their human-machine interface synchronized. Then, the MCU starts waiting for the new floor holder's audio packets.

A double synchronization mechanism was implemented to keep users' interfaces coherent: through the conference control protocol and through HTTP requests and responses. User indication in the interface is changed when the applet receives a control packet or an acknowledgement from the MCU. Each time a user indication is changed, the user submits to the web server his/her new indication. When a new user enters the conference room, his/her interface is loaded with updated information and he/she becomes aware of other participants in the conference room. Synchronization problems arise when control packets are lost or arrive out of order and when server information become inconsistent, but synchronization is restored during operation or by an explicit synchronization request submitted to the web server.

The double synchronization mechanism could be replaced by a reliable conference control protocol delivery mechanism or with explicit synchronization packets sent periodically from the MCU to the participants informing who is in the conference and what is his/hers current indication.

4 EXPERIMENTAL RESULTS

Some experiments were made in order to evaluate the quality of service parameters listed in Section 2: usability, interactivity, voice subjective quality and bandwidth needed for the system user and bandwidth, processing costs and system scalability for the system owner. In the first one, the floor control will be tested as a mechanism for restricting bandwidth consumption in centralized conferences and the bandwidth needed by the user and by the company will be revealed. In the second one, delay and packet loss will be seen through the use of a real world scenario (remote tests), impacting the voice subjective quality and the interactivity. In the third experiment, the MCU's scalability will be tested, bringing information about the systems' bottlenecks. In the last one, a subjective experience will be exposed, revealing information about the usability of the conference system.

The codec used in all tests was the GSM, with three 20 ms audio frames being carried by each RTP packet. Each audio frame occupies 33 bytes, so, the size (*S*) of the resulting Ethernet frame is given by Equation (1).

$$\begin{aligned} S &= 14\text{ B (Ethernet header)} + 20\text{ B (IP header)} + 8\text{ B (UDP header)} + 12\text{ B (RTP header)} + \\ &\quad 3 \text{ x } 33\text{ B (GSM audio payload)} + 4\text{ B (Ethernet FCS)} \\ &= 157\text{ B} \end{aligned} \tag{1}$$

Because GSM produces 20 ms audio frames, there are 50 audio frames per second. As the RTP is carrying three GSM audio frames per packet, 50/3 RTP packets per second are generated, increasing the codec delay from 20 ms to 60 ms, but lowering the bandwidth consumed by each audio channel from 36.400,0 bps ($B_{1\,GSM}$) to 20.933,3 bps ($B_{3\,GSM}$), as shown in Equations (2) and (3), respectively, because of less protocol overhead.

$$B_{1GSM} = 91\text{ B} \cdot 8 \cdot 50 = 36.400{,}0\text{ bps} \tag{2}$$

$$B_{1GSM} = 157\text{ B} \cdot 8 \cdot \frac{50}{3} = 20.933{,}3\text{ bps} \tag{3}$$

4.1 Floor Control and the Bandwidth Consumption in the Conference System

The first experiments were run in an Ethernet network and the bandwidth measurement was taken at the MCU, logging inbound and outbound audio RTP packets (RTP_RX and RTP_TX, respectively).

Two tests were made in order to show the floor control impact over the network bandwidth in a centralized conference. Test #1 (Figure 3, graph *A*) is a conference with two participants, while test #2 (Figure 4, graph *B*) is a conference with three participants.

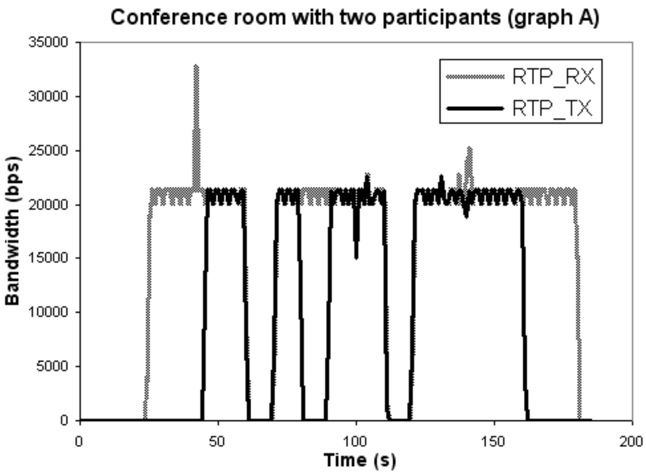

Figure 3 – Network bandwidth measured at the multipoint control unit (MCU) in test #1.

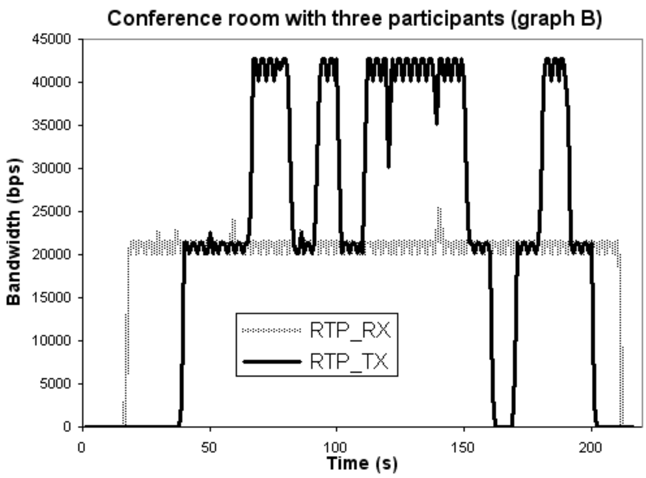

Figure 4 – Network bandwidth measured at the multipoint control unit (MCU) in test #2.

In test #1, when the moderator enters the conference room (around 20 s), he/she gains the floor and transmits its audio to the MCU. The MCU outgoing audio traffic is null, because there is no one to forward the packets to. When the second participant enters the conference (around 40 s), he/she starts transmitting his/her audio packets, because of NAPT issues, increasing the MCU's audio downstream until the participant receives a notification to stop transmitting. Although two audio streams are received by the MCU momentarily, the MCU only forwards the audio stream of the participant holding the floor, avoiding the increase of the audio upstream bandwidth. After the participant is admitted in the conference, the MCU starts forwarding the moderator's audio packets to him/her. When the moderator blocks his/her audio transmission by muting his/her microphone in his/her user interface (around 60 s), the MCU's audio downstream becomes null and the audio upstream too, consequently, because there are no packets to be forwarded. When the moderator blocks the participant's audio (around 80 s), then the MCU's audio upstream becomes null again. When the participant leaves the conference (around 160 s), the MCU's audio upstream becomes null again.

When the moderator is holding the floor and grants it to a participant (around 100 s in graph *A* and around 120 s in graph *B*), the MCU responds the floor granting with an acknowledgement control packet causing the moderator to stop transmitting. From this time on, the MCU will only forward the new floor holder audio stream, which has not yet starting transmitting it. After sending the ACKnowledge, the MCU forwards the floor granting to all participants in the conference. For a short period of time no one is transmitting, because the moderator has already stopped his/her transmitter and the participant to whom the floor was granted has not yet started his/her. This explains the drop in both the audio upstream and the audio downstream.

When a participant is holding the floor and the moderator grants it to another one (around 140 s in graph *A* and around 200 s in graph *B*), the MCU receives the floor granting, responds with an acknowledgement packet and starts waiting for the audio stream of the new floor holder, who has not yet started transmitting. For a short period of time, both participants are transmitting their audio, until the previous floor holder receives and processes the notification to cease his/her transmission. This explains the drop in the audio upstream and the increase in the audio downstream.

The main difference between test #1 and #2 is the audio upstream bandwidth, because of the third participant. Both tests show that floor control restricts the MCU's audio downstream bandwidth to the bandwidth occupied by a single audio channel, except for short period of times due to NAPT issues and floor transitions. Graphs *A* and *B* results just confirmed Equation (3) calculations on bandwidth.

4.2 Remote Tests

Three remote tests were made in order to evaluate the impact of a more realistic scenario in the conference system: test #1, using a broadband cable modem Internet service provider (provider *X*); test #2 and #3 using two different popular free dial-up Internet service providers (provider *Y* and *Z*, respectively), trough an analog modem.

In test #1 (Figure 5, network topology *A*), remote users A and B are located behind the same NAPT, while the server S is located in a different network. The equipment used as NAPT was a SMC Barricade 7004 ABR. Access to the Internet was provided by a Terayon TeraPro cable modem, connected at 256 kbps. In tests #2 and #3 (Figure 6, network topology *B*), user *A* is the remote user, while user *B* and server *S* belong to the same network.

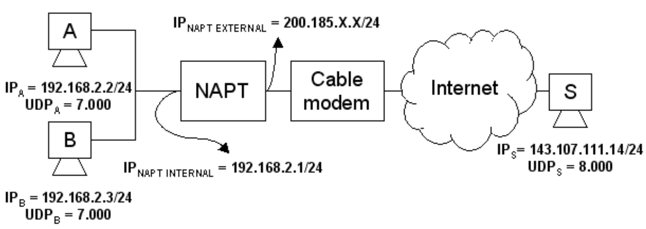

Figure 5 – Network topology *A*: remote test with a cable modem

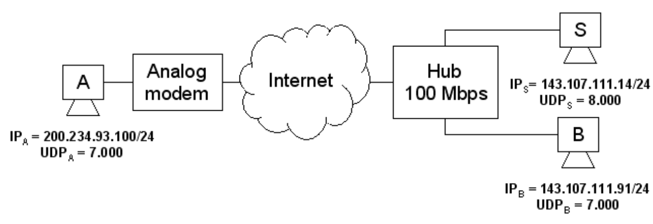

Figure 6 – Network topology *B*: remote test with an analog modem

Measurements were made using simple ICMP Echo Requests with 32 bytes of ICMP data transmitted from server *S* (IP address 143.107.111.14) to the external NAPT interface, in test #1 and directly to the remote user, in tests #2 and #3. The distance in hops between the networks were measured by sending ICMP Echo Requests setting the IPv4 *Time to Live* field (TTL) until ICMP Echo Replies were received instead of ICMP TTL Expired In-Transit messages. The impact of traffic prioritization was tested setting the *Type of Service* field (TOS). It was not possible to accomplish a trace route test in order to determine the exact path between server S and the remote hosts in each situation, because the intermediate hops did not respond to the ICMP Echo Requests. Table 3 presents the results of the tests.

Table 3. Measurements taken in the remote tests.

Test	IP address	Technology and connection speed	TOS	Distance in hops	Packet Loss (%)	Round-trip time (ms) Min/Avg/Max	Subjective evaluation
#1	200.185.X.X (full address omitted)	Cable modem @ 256 kbps (provider *X*)	0	13	6.5	20/40/110	No perceived delay or packet loss.
			224	13	12.5	20/41/120	
#2	200.234.93.100	Dial-up @ 52,0 kbps (provider *Y*)	0	12	56.5	761/827/841	High delay and packet loss.
			224	12	71.2	180/189/240	
#3	200.207.179.98	Dial-up @ 50,6 kbps (provider *Z*)	0	10	0.0	160/170/183	No perceived delay or packet loss.
			224	10	0.0	160/166/181	

The tests show that the path from server *S* to provider's *Y* network is very clogged (high delay and high packet loss rate), while the paths to providers' networks *X* and *Z* may not be congested. This explains why traffic prioritization is only perceived in test #2. Traffic prioritization seems to have a negative impact on tests #1 and #2, causing the raise of the packet loss rate.

As the conference system still does not implement traffic prioritization, the overall experience in test #2 was definitely unsatisfactory, mainly due to the high packet loss rate and not due to the extremely high delay. Tests #1 and #3 were very satisfactory. Packet loss caused short interruptions in the audio and some non-critical synchronization problems in test #1. Users could not perceive any differences in test #1 and test #3, regarding the interactive experience, despite test #3 higher delay.

The conference system worked flawlessly and transparently with the NAPT in test #1. Audio quality was normal in tests #1 and #3. In terms of responsiveness, the interactivity was excellent in tests #1 and #3, but test #2 was nearly impracticable due to packet loss and high delay.

4.3 Scalability Test

Scalabilty for the conference system is related to the maximum number of participants that can take part simultaneously in conferences and is mainly affected by the MCU's available network bandwidth and by the MCU's performance. Equation (3) shows that the codec used requires around 21 kbps per participant and the tests in Section 4.1 reveal that the MCU's uplink bandwidth grows linearly with the number of participants.

A virtual conference room with ten participants was used in this test, all of them belonging to the same local network (full-duplex Fast Ethernet) as the server, which was configured with a Pentium III 600 MHz processor and 512 MB of RAM.

Two response time were measured in order to evaluate the MCU's performance: the time taken by the MCU to start forwarding a valid RTP packet from the floor holder to the rest of the participants (T_{first}) and the time between forwarded packets (T_{next}).

Table 4 shows the packets captured by a sniffer software and the delta time between them. Packets #1, #11, #21 and #31 are GSM packets received by the MCU from the floor holder. Packets #2 to #10, #12 to #19 and #22 to #29 are GSM packets forwarded to the rest of the participants. Although the MCU should receive a GSM packet each 60 ms (*D*), there is a very high jitter caused by the JMF transmitter.

Table 4. Delta times of packets received and sent by the MCU.

	Floor holder $\rightarrow$ MCU	MCU $\rightarrow$ participants								
Packet	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Delta time (μs)	0	453	160	143	146	142	145	145	145	146
Packet	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
Delta time (μs)	87.973	447	162	144	146	143	146	145	144	148
Packet	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30
Delta time (μs)	64.468	389	160	144	147	145	145	144	145	146

After receiving a valid RTP packet from the floor holder, the MCU takes approximately 450 μs to forward the first GSM packet. This time is spent by the reception routine, validating the RTP header, determining if the packet belongs to the floor holder and if the UDP source port must be captured. After validation, the transmission routine verifies if the participant is allowed or is ready to receive the audio stream, changes the RTP header, the destination IP address, the destination UDP port and finally forwards the GSM packet to him/her. After the first GSM packet is forwarded, the MCU takes approximately 160 μs to forward the second one and approximately 145 μs to forward the next ones. This time is consumed by the MCU iterating through its participant's data structure and in the transmission routine.

Assuming T_{first} as 500 μs, T_{next} as 160 μs, the approximately number of participants (N_{served}) that can be served by the MCU is given by Equation (4) below.

$$N_{served} = \frac{D - T_{first}}{T_{next}} = \frac{(60\,\text{ms - }0{,}500\,\text{ms})}{0{,}160\,\text{ms}} = 371\,\text{participants} \tag{4}$$

A virtual conference room with 371 participants would require an MCU with almost 10 Mbps of uplink. A room with such a high number of participants would be extremely difficult to be managed by the moderator. A more realistic scenario would be simultaneous conferences with a much smaller number of participants in each room. For each virtual conference room, the MCU receives one audio channel. For each RTP packet received, T_{first} must be considered. Adapting Equation (4) in order to consider m simultaneous conferences taking place produces Equation (5).

$$N_{served}(m) = \frac{D - m \cdot T_{first}}{T_{next}} \tag{5}$$

As an example, the conference system hosting 10 simultaneous conferences would be able to serve 346 participants, approximately 35 participants per conference.

Tests made with two and three virtual conference rooms with five participants each showed no change in the response times. This test has shown that the MCU's bandwidth will frequently be the main bottleneck of the conference system. Besides, response time may be improved with higher clock speed processors and the MCUs could be distributed among servers over the network.

4.4 Subjective Evaluation

The floor holder has a very strange silence feeling, as he/she does not receive his/hers own audio. The audio cue helps the participant to recognize when he/she receives the floor, mainly when his/her attention is needed in different parts of the screen. Due to the lack of visual contact, the new floor holder uses to test the communication by asking "Are you hearing me?" or "If you hear me, send a text message!", turning the text communication into a feedback channel. The absence of visual communications also causes the participants to get more easily distracted.

Frequently, the floor holder forgets that his/hers question can only be answered if the floor is granted to the other participant. In turn, the listener many times forgets that his/hers words are not been heard by anyone and tries to interact with the floor holder without receiving the floor. When he/she receives the floor and hears the audio cue, he/she remembers that only now the other participants will be able to hear him/her.

Major communication problems arise when the participants try to use the conference system like in a phone call, when it was developed to reproduce a classroom, with an asymmetrical flow of information. However, when the system is used as a virtual classroom, with the teacher speaking most part of the time, after some practice all participants seem to get more comfortable with the floor control mechanism.

5 CONCLUSIONS AND FUTURE WORK

Floor control has proved to be an efficient mechanism for restricting bandwidth consumption in centralized conferences. Further developments in the floor control protocol will seek robust synchronization for the participants and the abstraction of the resource type. When the conference system is used as a virtual classroom, the floor control mechanism shows itself appropriated for regulating the conversational flow.

The greatest strengths of the conference system developed are: the utilization of web-based technology, easing the integration with other systems and the customization of the human-machine interface; transparent NAT/NAPT operation; the use of floor control as a mechanism to restrict network bandwidth in conferences over IP and the codec independent MCU.

The absence of a proper return channel and the lack of mutual awareness are critical for the conversational flow and for raising the degree of interaction. The use of video would certainly help. Sending the mixed audio of all participants to the floor holder would be very processor intensive, would make the MCU codec dependent and would considerably increase the bandwidth consumed by it, which is already the biggest bottleneck for the conference system's scalability.

Further studies are needed in order to find out a more efficient way to provide mutual awareness and fast feedback from the participants, without impacting significantly the consumed bandwidth. Qiu, Kuhns and Cox (2002) propose a solution to raise interactivity in decentralized conferences that could be adapted to centralized ones, with the MCU forwarding the audio from the participant determined by the moderator to the floor holder.

ACKNOWLEDGEMENTS

This work is part of the Voice Chat project supported by Scopus Tecnologia S. A. <URL: http://www.scopus.com.br>.

REFERENCES

CROWLEY, T., MILAZZO, P., BAKER, E., FORSDICK, H., TOMLINSON, R. 1990. MMConf: An infrastructure for building shared multimedia applications. In: *Proceedings CSCW'90*, p. 637-50, Oct.

DOMMEL, H.-P., GARCIA-LUNA-ACEVES, J. J. 1997. Floor Control for Multimedia Conferencing and Collaboration. Multimedia Systems, vol. 5, no. 1, p. 23-38.

HOLDREGE, M., SRISURESH, P. 1999. *IP Network Address Translator (NAT) Terminology and Considerations*. RFC 2663, IETF, Aug. Available from web: <URL: http://www.ietf.org/rfc/rfc2663.txt>

HOLDREGE, M., SRISURESH, P. 2001. *Protocol Complications with the IP Network Address Translator*. RFC 3027, IETF, Jan. Available from web: <URL: http://www.ietf.org/rfc/rfc3027.txt>

QIU, R., KUHNS, F. COX, J. 2002. A Conference Control Protocol for Highly Interactive Video-conferencing. In: *Proceedings of Globecom 2002*. Taipei, Taiwan, Nov.

SCHULZRINNE, H. ET AL. 1996. *RTP: A Transport Protocol for Real-Time Applications*. RFC 1889, IETF, Jan. Available from web: <URL: http://www.ietf.org/rfc/rfc1889.txt>

SENIE, D. 2002 *Network Address Translator (NAT) – Friendly Application Design Guidelines*. RFC 3235, IETF, Jan. Available from web: <URL: http://www.ietf.org/rfc/rfc3235.txt>

SMETANA, G. ET AL. 2004. A floor-controlled centralized conference system for distance learning. In: *Proceedings of The IASTED International Conference on Web-Based Education*. Innsbruck, Austria: International Association of Science and Technology for Development (IASTED), 2004, 5p.