Automatic Addition and Deletion of clients in VoIP Conferencing

R Venkatesha Prasad*, Joy Kuri*, H S Jamadagni*, Haresh Dagale*,
Ravi A Ravindranath+
*Centre for Electronic Design and Technology, Indian Institute of Science, Bangalore, India.
+ Presently at ADC Telecommunications Inc., Mountain View, CA.
E-mail: {vprasad, kuri, hsjam, haresh}@cedt.iisc.ernet.in

Abstract[1]

In a conference, considering the packets only from a set of selected clients can reduce the degradation of the quality of speech because mixing packets from all clients can lead to lack of clarity in the speech of any participants. The automatic selection should be smooth and should not cause frequent interruptions. A method of selecting the clients for mixing is suggested here based on a new quantifier of the voice activity called Loudness Number (LN). The dependence of the Loudness Number on the amplitude of the packet at present and the past activity is clearly brought out. The structure of the packet used has been explained. A method to avoid echo and enhance the quality of the conference is presented. The contributions of the paper are expected to aid in the implementation of H.323 recommendations for the Multipoint Processors (MP). A working prototype based on the proposed Loudness Number is already functional.

Keywords: Loudness Number, Call Processor, Selector

1. Introduction

Today's Internet uses the IP suite, which offers best effort data delivery and was primarily designed for transport of data. Increasingly however, the Internet is being used as a transport mechanism for voice and video, which have different characteristics and requirements from those of traditional data. For "telephone conversation-type" applications, there are strict requirements on end-to-end delay and delay jitter. There are a few studies that examine the efficiency and quality of packetized voice [2]. It is found that due to statistical multiplexing and compression schemes, the efficiency in terms of the volume of voice traffic carried can be increased to a large extent. There are also some studies about the effects of Codecs (vide [7] for Codecs) and QoS guarantees in [2] and [9].

[1] This work was supported by Nortel Networks agreement number RIISOG9900HSJ

The next step in the process of merging telephony with the Internet is providing a number of facilities that a telephone network provides. Among them, the conference facility is the most important. The reasons for the popularity of audio or video conferencing on the Internet are dealt with in detail in [1]; the advantages of audio and video conferencing have been thoroughly explored in [3] and [10]. In its simplest implementation, the bandwidth requirement for a conference over the Internet is directly proportional to the number of participants (a participant is also referred to as a "client"). Reducing bandwidth for conferencing while maintaining audio quality is a challenge in Internet Telephony. Apart from bandwidth, the other issues are: (a) packet delay, (b) echo suppression, (c) mixing of audio from selected participants, (d) automatic selection of participants for mixing, (e) playing of mixed audio at each participant, (f) handling participants not capable of mixing audio streams (such participants are known as "dumb participants"), and (g) deciding the number of participants that can be in conference without degrading voice quality.

This paper explores a technique for seamless addition and deletion of clients whose packets have to be mixed for play out at each client in the conference. In section 2 the ITU-T H.323 recommendations are considered. Section 3 has a brief description of the new parameter called the loudness number, based on which automatic selection of clients in a conference is implemented. Section 4 describes a simple algorithm and an example for automatic selection of audio streams based on "Loudness Number". Implementation of the entire system is given in section 5 and conclusions are in the section 6.

2. H.323 standards on Audio conferencing

H.323 [5] defines multimedia communication standards for the IP-based networks. Designed to compensate for the effect of highly variable LAN latency, H.323 allows customers to use multimedia applications without changing their network infrastructure and ensures interoperability between H.323 compliant products. H.323 defines Multipoint Control Unit (MCU) and Terminals,

which are the key elements in the architecture. The Multipoint Control Unit (MCU) is an endpoint on the network, which provides the capability for three or more terminals and Gateways to participate in a multipoint conference. The MCU consists of a mandatory Multipoint Controller (MC), and optional Multipoint Processors (MP). Communication between the MC and the MP is not subject to standardization.

2.1 Multipoint Controller (MC)

The MC performs H.245 [6] multipoint control functions to support a multi-point conference. The MC carries out the capability exchange between endpoints in a multipoint conference. It ensures common operating mode. Clients joining or leaving a conference may change common operating mode of conference. The MC does not deal directly with any of the media streams. Communications between MC and MP are not subject to standardized.

2.2 Multipoint Processor (MP)

The MP receives audio, video and/or data streams from the endpoints involved in a centralized or hybrid multipoint conference. It processes these media streams and returns them to the endpoints. The MP may process one or more media stream types. An MP that processes audio shall prepare N audio outputs from M audio input streams by selecting, mixing, or a combination of these (Figure 1). Audio mixing requires decoding the input audio to linear signals (PCM), performing a linear combination of the signals and re-encoding the result to the appropriate audio format. The MP may eliminate or attenuate some of the input signals in order to reduce noise and other unwanted signals. Each audio output may have a different mix of input signals providing for private conversations. The terminals shall assume that their own audio packets are not present in the audio stream returned to them. Mixing in a terminal makes it possible to suppress the terminal's own audio stream and therefore avoids echo.

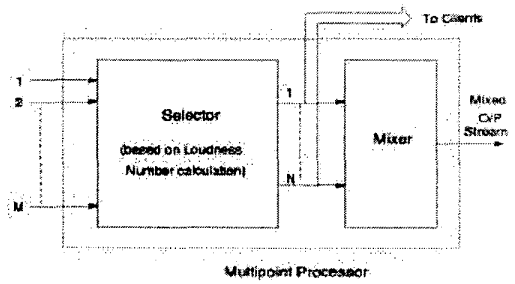


Figure 1. Architecture of Multipoint Processor

3. Loudness Number

A basic problem to be solved by the MP is: in a mixing interval, how should it choose N packets out of the M it might possibly receive? One way to do this would be to rank the M packets received according to their energies, and choose the top N. However, this is usually found to be inadequate because random fluctuations in packet energies can lead to poor quality audio. For example, sudden noises in a listeners environment can cause a transient spikes in packet energy, leading to that noisy packet being chosen among the N instead of a legitimate speaker's packet. This indicates the need for a metric different from mere individual packet energies. The metric should have the following characteristics:

- A person who is speaking (i.e., "has the floor") should not be easily cut off by transient spikes in packet energies of the other participants. This implies that a speaker should have some "weight" depending on his/her past activity; this weight is often referred to as "Persistence" or "Hangover".
- By the same token, a participant who wants to interrupt the speaker will have to (i) raise his voice and (ii) keep trying for a little while in order to break in. In a real-life conference, the body language of a participant often indicates that he wants to interrupt. But in the audio conferencing scenario under discussion, a participant's intention to interrupt can only be conveyed through the loudness metric on the basis of which the packets to be mixed are selected.

To satisfy the above requirements, we define new a metric called Loudness Number, which changes slowly with time so that the selection (addition and deletion) of clients for conference is smooth.

The Loudness Number (λ) is a function of the amplitude of the audio stream, amplitude during a well defined past duration, as well as voice (speech) activity of the speaker for a well defined past. Decision to add and delete clients to conference will be based on this Loudness Number.

3.1. Definition of Loudness Number

The Loudness Number is calculated on per packet basis. The basic parameter used in this calculation is packet amplitude, which is calculated as root mean square (rms) of the energies in audio samples of a packet, and is denoted by X_K. We define three windows, as shown in Figure 2 for calculating the loudness number. These three windows contribute to the three parameters that are to be considered while calculating the loudness number.

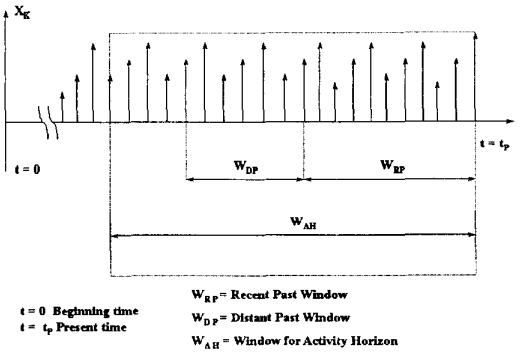


Figure 2. The Different Windows for Loudness Number calculation

The present amplitude level of the speaker is found by calculating the moving average of packet amplitude (X_K) within a window called "*Recent Past Window*" starting from the present instant to some past time. The past activity of the speaker is found by calculating the moving average of the packet amplitude (X_K) within a window called "*Distant Past Window*", which starts at the point where the "Recent Past" window ends and stretches back in the past for a pre-defined interval. The activity of the speaker in the past is found with a window called "*Activity Horizon*", which spans the recent past window as well as the distant past window and beyond if necessary. Though the contribution of the activity horizon looks similar to the contribution of the recent past and distant past windows, past activity is calculated from activity horizon window in a different way.

We define the quantities during these three intervals as L_1, L_2 and L_3. L_1 quantifies the Recent Past speech activity, L_2 the Distant Past speech activity and L_3 gives a number corresponding to the speech activity in the Activity Horizon window quantifying how active the speaker was in the past few intervals. L_3 yields a quantity that is proportional to the fraction of packets having energies above a pre-defined threshold (equation 3 below). The threshold is the same for all the clients.

Loudness Number λ is defined as a convex sum of L_1, L_2 and L_3 with weighting factors α_1, α_2 and α_3. The definitions for L_1, L_2 and L_3 are as shown below.

$$L_1 = \frac{1}{W_{RP}} \sum_{K=t_P}^{t_P - W_{RP}+1} X_K \qquad (1)$$

$$L_2 = \frac{1}{W_{DP}} \sum_{K=t_P - W_{RP}}^{t_P - W_{RP} - W_{DP}+1} X_K \qquad (2)$$

$$L_3 = \frac{1}{W_{AH}} \sum_{K=t_P}^{t_P - W_{AH}+1} \theta * I_{\{X_K \geq \theta\}} \qquad (3)$$

Where, $I_{\{X_K \geq \theta\}} = 1 \; if \; X_K \geq \theta$

$= 0, Otherwise$

The threshold θ is a constant. We have set θ at 10-20 percent of the amplitude of the voice samples of a packet in our implementation. Now the Loudness Number λ_p for the present time instant (or the present packet) is calculated as,

$$\lambda_{tp} = \alpha_1 * L_1 + \alpha_2 * L_2 + \alpha_3 * L_3 \qquad (4)$$

Here the α_1, α_2 and α_3 are chosen such that,

$0 < \alpha_1, \alpha_2 < 1, \; 0 < \alpha_1 + \alpha_2 < 1$, and $\alpha_3 = 1 - (\alpha_1 + \alpha_2)$.

Here, α_1 is the weight given to the recent past speech, α_2 is the weight given to distant past speech and α_3 is the weight given to speech activity in the activity horizon window considered. The Loudness Number varies slowly because of the moving averaging used for its calculation. By choosing appropriate values for the lengths of the windows, α_1, α_2, α_3 and θ can be adjusted so that a client can be added or deleted from a conference smoothly.

3.2. Computational Complexity of Loudness Number

The computation complexity of the Loudness Number calculation does not depend on the window size (after an initial transient period) as we can implement it using a circular buffer. With X_K, the rms value of a packet, being available, the total number of additions and multiplications are 17 and 9 respectively for calculating the loudness number. Thus the calculation of loudness number is simply implemented in real time with the only requirement of circular buffer management. The reduction in number of additions and multiplications comes from the fact that at each instant the outgoing packet amplitude X_K is subtracted and the new one is added to the sum. This can be done with the help of pointers pointing to the oldest sample in a circular buffer.

4. An example of the VoIP Conference with Loudness Number

The architecture for conferencing is shown in Figure 3 and Figure 4. Functionally, Call Processor (CP) is similar to MC and Selector is similar to the MP of H.323. There may be many Selectors supporting a conference and Selectors are required to register with CP.

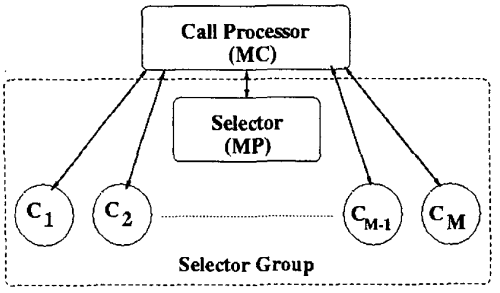


Figure 3. Control Path Structure of VoIP Conference

Now we shall consider the functions of the CP and Selectors. The Signalling (on the control path) is required between clients and CP as well as CP and Selector. The CP implements all control messages and is responsible for setting up the conferences. A client will be invited into a conference by another client/moderator of the conference or will join an existing conference. If a client joins the conference CP informs all other clients and the Selector about the arrival of the new client. The Selector will serve the client that has joined the conference. The CP also decides N, the number of clients to be selected for play out amongst all the clients in a conference communicates to the Selector.

The diagram in Figure 4 shows the data flow between the Clients and Selectors and Selector to Client. The Selector to client communication can be over unicast or multicast. As most of the LANs support multicasting, a Selector can usually talk to its clients on multicast. But in cases where the multicast support is not available on the LAN, the Selector has to send unicast packet to that set of clients, as shown in Figure 4. The Selector sends a mixed audio packet in case a set of clients cannot mix the audio streams themselves.

Selectors are used only for conferencing. Selectors handle only the audio streams and can also convert audio stream formats if necessary. A Selector picks the first N packets out of M clients based on highest Loudness Number. These N clients along with their identification ID (usually the IP# / E-mail) form a set say *S* whose audio packets are to be mixed and played out at each client. The system also has the following important advantages:

(a) N clients in the conference are decided based on the Loudness Number of each client. Use of loudness numbers to select participants in a conference reflects the real-life face-to-face conferences wherein participants try to get (cut) into the conference by force.

(b) Clients mix the N packets sent by the Selector based on the weights set by the user for each of the selected N clients' audio through a GUI. The quality of

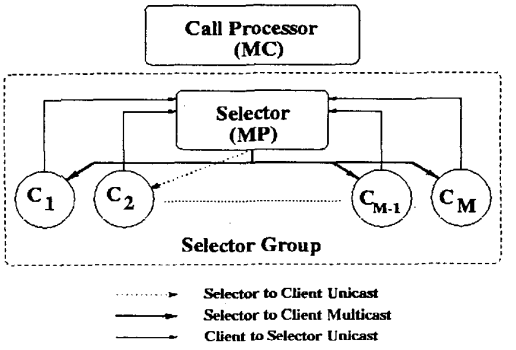


Figure 4. Data (Voice) Path Structure of VoIP Conference

the conference is enhanced appreciably as each user can adjust the volume of the speaker to whom the user likes to listen attentively.

(c) The N clients, which are selected will get their own packets back. Mixing is done based on the weights set by the user for each of the N clients. While mixing the weight for the "self" can be set to a very minimal value so that the echo of his/her own voice is not heard. This ensures that the user can know that his voice packets are selected and is heard by all other clients without affecting the quality of mixed speech at his terminal.

(d) This architecture can be extended with multiple Selectors forming multiple Selector groups. As each Selector from a group will be sending only N out of M packets to the other Selectors (M > N), bandwidth used by the application over a wide area network is bounded above by N. The Selector will have to handle only N*(P-1) number of packets (P = number of Selectors) from other Selectors and not M*(P-1). This saves bandwidth and saves computation at each Selector, and leads to a scalable architecture with multiple Selectors.

4.1. Automatic Selection of clients

In Figure 4, out of M clients in the conference only four (N=4) will be selected at any point of time in the conference. Only these four clients' audio streams will be mixed and played out. We shall consider briefly the format (Figure 5) of the packets, which flow from a client to the Selector and Selector to clients. This format is a an extension of RTP [4]. We have proposed a few extra fields to support new facilities and this may be accommodated later into the existing RTP packet format. The packets from the clients have all the usual RTP fields plus a field for Loudness Number and packets from Selector to clients have sets of Contributing Source ID, Loudness Number and the Data from the selected client.

An example of a packet from a Selector to its clients with N = 2 is shown in Figure 5. The packetization is done for every 40ms, depending on whether 8 or 16 bit PCM encoding is used, the data length is 320 or 640 bytes respectively.

The algorithm for identifying the set S is simple and self-explanatory. It runs at Selector,

Repeat for each time slot at Selector
{

1. *Get all the packets from the clients.*
2. *Out of M, select the N clients with the highest Loudness Numbers.*
3. *Put these N packets in a buffer (set S).*
4. *If N is large then split the set S into smaller groups so that the packets in each group can be sent in chunks.*
5. *Send the packets in each subsets of S to the clients on Multicast (Unicast if the client cannot listen on Multicast address).*
6. *Mix these N audio packets in set S after linearising and send it to the dumb clients.*

}

The set C denotes the set of clients with Loudness Number at a Selector, the Loudness Number is in bracket adjacent to the client number, for example, C = {1(34), 2(45)} means the set C has two clients 1 and 2 and their respective Loudness Numbers are 34 and 45

Now we shall take an example of a conference scenario with 10 clients (M= 10). The set C given below is for one time slot at any instant of time.

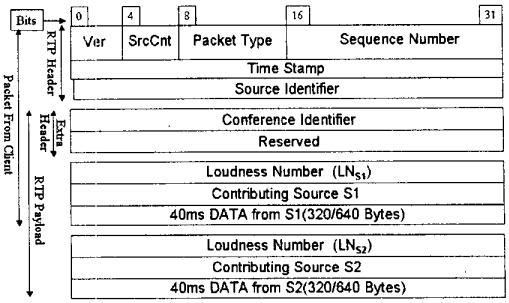
C = {1(80), 2(91), 3(22), 4(23), 5(24), 6(25), 7(35), 8(21), 9(20), 10(21)}

Now Selector will select the best N (in this case N = 4) packets from its set C and forms the set S. For this example the set S is,

S = {1(80), 2(91), 7(35), 6(25)}

The packets from the set S will be mixed and played out at the clients with appropriate weights for each stream as set by the client. In case of dumb terminals the Selector mixes the packets in the set S with predefined weights and send a single packet to dumb terminals.

All participants in the conference, i.e., clients 1 to 10, will listen only to the clients in the set S; clients number 1, 2, 7, and 6 during the current time slot. The same selection process is repeated for each time slot till the end of the conference thus dropping clients with less λ_{tp}. The set C will grow whenever a new client is allocated to that Selector group and will shrink when a client goes out of the conference. The cardinality of the set S might be less than N if the corresponding set C has less than N members.



Note: N = 2; for N more than 2, the RTP Payload will have more sets of LNs, Contributing Sources, and DATA

Figure 5. UDP Packet Structure (Client to Selector and Selector to Client)

5. Implementation

The Call Processor and Selectors are implemented on Windows NT workstations. The Call Processor builds the conference, and all the selection and switching of the voice packets from the clients is done at the Selector. Call Processor does a subset of H.323 call/conference implementation. The client program can run on any of the Windows 95/98/NT/2000 operating systems. The clients have graphical user interface for setting up the conference and changing the weight for mixing for each of the clients that are in the set S. The set up of the conference is as shown in the Figures 3 and 4 which gives the control and data flow in a typical conference.

5.1. The Call Processor (CP)

The Call Processor handles all the control messages to set up the call/conferences. The clients will have to register with the CP before they can set up a call/conference with other clients. The CP opens up a terminal object whenever a client registers and stores the IP number and the E-mail address of the user. If a client wants to connect to another client, it has to send a request (equivalent to dialing the digits) specifying the E-mail address or the IP number of the called party. The CP will create a call object corresponding to this request. The CP will contact the called client and a ring tone would be sent. After getting an affirmative response, both the clients are given the UDP port on which they should send and receive UDP packets. When a client in conversation (a two party call) requests to add the third party thereby converting the call to a conference, the CP will instruct the Selector to receive the UDP voice packets from all the clients in the conference. The Selector and all the clients are also instructed by CP to send and receive (respectively) on a multicast IP number. All the control

signals between clients and Selector are routed through the CP and a TCP connection is used for reliable transmission of these control signals at present.

The CP is implemented using object oriented software design. The calls and conferences are different objects derived from the call/conference class. The CP sets up calls based on Email addresses or the IP numbers. Call set up based on Email results in the characteristic that calls to the user would automatically follow the user wherever the user is. CP can run on any Windows platform.

5.2. Selectors

The Selectors are implemented on Win NT computers because of the higher accuracy of the timer routines as Win 95/98 OS cannot satisfy the requirement of accurate timer messages. The Selector is also designed using object oriented software design. A conference object contains objects and data structures for each client in that conference. Packets from each client are arranged in a queue based on the timestamps of the packets. The packets from the head of the queue of all the clients are considered for selection at every mixing interval. The format of the audio packets for each client may be different.

5.3. clients

The clients are implemented using Microsoft Foundation Class (MFC) support for Windows programming. The client program can run on Intel processor based Win95/98/NT/2000 computer. The user has to register with the CP in the beginning with the E-mail address and /or IP number. The GUI provides all the required facilities to connect to another client and getting into a conference. The user has the facility of setting the weights for each active client at any time. The values of weights determine the volume of the speakers at the mixer output. If the audio packets are coded/compressed then it has to linearized before mixing. Whenever the user is one of the active clients selected by Selector, echo can be avoided by assigning a small weight to "self" packet.

The Loudness number is calculated at each client and it is sent along with audio data using RTP. For calculating the LN the values for the Recent Past window, Distant Past window and the Activity Horizon are 10 seconds, 15 seconds and 30 seconds respectively. The values for α_1, α_2 and α_3 are 0.4, 0.3 and 0.3 respectively.

6. Conclusions

This architecture avoids impulse sounds because of the way the loudness number is implemented. A persistent speaker gets into the conference i.e., into the set *S*. This set up models closely what happens in a typical face-to-face conference wherein there is always a chance for others getting into the addressing mode along with the persons who is speaking at that instant.

A design choice of N = 4 has worked well in our tests because in any normal conference, the number of persons speaking at any given instant of time will be only one, but in some cases when other members of the conference interrupt, it may be two, three or at most four.

The facility of user-specifiable weights for mixing gives an opportunity for boosting the voice of the client whom the user wants to hear most clearly, as well as a simple way for a speaker to avoid listening to his own voice in the mix (echo suppression). This is achieved at the cost of Selectors sending N packets instead of a single mixed packet. However, given current LAN technology, the increased bandwidth requirement should pose no problem at all.

Some more facilities can be easily added here: one of the clients can be the moderator and that client will be given priority. We can also give the responsibility to the moderator for allowing one of the clients to be in the set of N clients. The client thus selected by the moderator will not have to compete.

References

[1] Lisa R. Silverman, "Coming of Age: Conferencing Solutions Cut Corporate Costs" White Paper, http://www.imcca.org/wpcomingofage.asp

[2] Mario Baldi and Fulvio Risso, "Efficiency of Packet Voice with Deterministic Delay", *IEEE Comm. Magazine*, May 2000, pp. 170-175.

[3] Maurizo Decina and Vittorio Trecordi, "Voice over Internet Protocol and Human Assisted ECommerce" *IEEE Comm. Magazine*, Sept. 1999, pp. 64-67.

[4] H. Schulzrinne et al., "RTP: a transport protocol for real-time applications", RFC 1889, IETF, Jan. 1996, ftp://ftp.isi.edu/in-notes/rfc1889.txt

[5] ITU-T Rec. H.323, "Packet based Multimedia Communications Systems", vol. 2, 1998, http://www.itu.int/itudoc/itu-t/rec/h/h323.html.

[6] ITU-T Rec. H.245, "Infrastructure of audiovisual services – Communication procedures", http://www.itu.int/itudoc/itu-t/rec/h/h245.html.

[7] Robert Shaw, reference implementation for CCITT G.711, G.721, G.723, Jun. 1993 http://www.itu.int/itudoc/itu-t/rec/g/g700-799/ refimpl.txt.

[8] ITU-T Rec. T.120, "Data protocols for multimedia conferencing", http://www.itu.int/itudoc/itu-t/rec/t/t120.html.

[9] Markku Korpi and Vineet Kumar, "Supplimentary Services in the H.323 IP Telephony Network", *IEEE Comm. Magazine*, July 1999, pp. 118-125.

[10] Amitava Dutta-Roy, "Virtual Meetings with desktop conferencing", IEEE Spectrum, July 1998, pp. 47-56.