An Internet Telephone Software System for Real-Time Voice Communication

K.V. Chin, S.C. Hui and S. Foo
Division of Software Systems, School of Applied Science
Nanyang Technological University, Singapore

Abstract

This paper describes the design and implementation of an Internet Telephone Software System which allows real-time full duplex voice communication between two parties through the Internet. The system consists of two main components: Telephone Exchange and Telephone Software. The Telephone Exchange manages a list of active users who are connected to the system, establishes the telephone connection and monitors the quality of services. The Telephone Software is responsible for controlling session and audio management processes. Session management is concerned with the negotiation of communication parameters between two parties before actual communication can take place. Audio management is responsible for audio transmission and the recording and playback operations of audio data. The system has been implemented on Unix-based Sun Sparc workstations using Sun audio cards at the School of Applied Science, Nanyang Technological University. This paper discusses a number of design issues in the development of Internet telephone systems and shows how these are considered in the implementation of the Internet Telephone Software System.

Keywords: Internet telephone systems, software systems, real-time voice communication

1. INTRODUCTION

Recently, as a result of declining costs of computer hardware, advances in computer technology and phenomenal growth in Internet, a number of research prototypes and commercial products of Internet telephone systems have been developed [1] to offer real-time voice communications and other value added services over the Internet. The growing enthusiasm stems mainly from huge potential cost savings by making it possible to make transcontinental telephone calls at the prices of local telephone calls plus nominal standard Internet connectivity charges.
Figure 1 shows the basic components of an Internet telephone system. Two host computers acting as caller and recipient are required. In using the standard Internet Transmission Control Protocol/Internet Protocol (TCP/IP) [2], each host computer is identified by a unique IP address. The host computer can either be a workstation or a personal computer with sufficient computation power and audio capabilities. The telephone software system which resides on each host computer facilitates the real-time voice communication across the Internet. In the basic communication process, the caller’s software system will acquire the real-time voice data through an audio input device and convert the analogue signals into digitized form which is then compressed and optionally encrypted before being transmitted to the recipient through the Internet using the TCP/IP protocol. Compression is necessary to reduce the bandwidth requirement of the voice data. At the recipient’s end, the software system carries out the reverse process. Incoming data is first decrypted, decompressed and played back in real-time on the audio device of recipient’s computer. Communication can either be half or full duplex although the second form is desired since it emulates the conventional telephone system.

However, the quality of communication using these products is still not comparable to those offered by telephone companies [3]. The inferior quality is mainly due to the high transmission delay and packets lost of the Internet environment which is characteristic of packet-switched network without resource reservations mechanisms. As illustrated in Figure 1, the quality of voice communication over the inter-networks depends on the roles played by the host computers, network equipment and network protocols used. Therefore, in order to achieve satisfactory real-time voice communications over the Internet which does not guarantee any form of quality of service, it is necessary to study the network aspect to establish how it can be more effectively utilised to support real-time communications. To carry out this study, we have developed an Internet Telephone Software System [4,5], which is capable to delivering full-duplex real-time voice communications between two parties across the Internet, at the School of Applied Science, Nanyang Technological University. The system aims at offering good performance in terms of hearing perception against the contradictory factor of low cost in terms of CPU and network load.
In this paper, we first discuss a number of key design issues in the development of Internet telephone systems. Subsequently, the architecture of the Internet Telephone Software System and its main components are presented. The communication process used in the system to establish a connection between a caller and recipient is then described. Finally, a summary of the work and directions for future work are given.

2. DESIGN ISSUES IN INTERNET TELEPHONE SYSTEMS

A number of design issues are identified for the development of an Internet telephone system. As almost all existing Internet telephone systems are developed with a commercial interest in mind, very little or no technical information and system description are available. These design issues are generally inferred from available literature and the end-result of studying and using these systems.

As shown in Figure 1, two possible modes of connection to the Internet are possible. In the direct connection, users are connected directly to the Internet and have fixed IP addresses. In the second form, users gain access to the Internet via an Internet Service Provider (ISP) (or Internet Access Provider (IAP)). In this instance, the IP address is dynamically allocated at connection time. This IP address will thus remain fixed during the current session. It will subsequently be reused for other users after the current user exits from Internet. Connecting to Internet in this manner will generally result in a different IP address being assigned each time. Connection using the first mode poses no problems since the IP addresses are fixed and known beforehand, so that a caller and recipient pair will only have to start the telephone software at their respective host computers and communication can take place. However, the dynamic IP addressing in the second mode poses special problems since the IP address is only known during connection time. As this mode of connection would affect the majority of users on Internet, there is a need to identify a suitable method to resolve this dynamic IP addressing problem before communication can take place.

The TCP/IP protocol provides two kinds of network services to application processes: Transport Control Protocol (TCP) which is a connection-oriented protocol with guaranteed delivery of data and User Datagram Protocol (UDP) which is a connectionless-oriented protocol with no guarantee of arrival of data. The protocol applies well for non-temporal data transmission but does not have any provisions to support the real-time nature of the audio. The Internet telephone system is thus constrained by this major limitation in delivering real-time service. Additional information has to be supplied to describe and construct the real-time nature of this service.

As the Internet telephone system is to operate in real-time, two possibilities exist. The first is to develop a new real-time protocol or modify and extend the existing TCP/IP to support real-time communications. This new protocol must be backward compatible to support existing TCP/IP based applications. Such an approach of guaranteed real-time delivery of data will transform Internet to support synchronous communication so that live audio and video conferencing, real-time services-on-demand and other forms of real-time groupware applications become possible. This development is costly and complex and is subjected to widespread adoption to make it into a future standard. The second possibility is to make use of the existing TCP/IP protocol and use new mechanisms and descriptors to extend the protocol to deliver real-time service. This approach will at best simulate real-time but does not guarantee real-time delivery due to the underlying nature
of the existing protocol. This second approach is basically adopted by all existing Internet telephone systems to deliver real-time audio communication.

The quality of the Internet telephone conversation should be compatible to that of a conventional telephone. This is a very difficult criterion to achieve as the telephone software system is at the mercy of the Internet transmission medium to deliver the audio packets in real-time to enable such quality to be achieved. In general, it is not economical to transfer raw Pulse Code Modulation (PCM) audio signals as this will severely burden the network to support yet another form of audio data. Conventional telephone systems which have a bandwidth of 300 Hz to 3400 Hz will translate into a need to digitise voice samples at 8000 Hz [6]. With 8 bits per sample, this will amount to 64 Kbps. This enormous amount of data transmission will put the network under severe stress. At the same time, it cannot be supported using existing 28.8 Kbps modems.

One possible solution is to employ efficient encoding algorithms to transform analogue voice into digital signals. Many such algorithms are already in existence [2]. Compression is clearly necessary to reduce the bandwidth requirement of these digitised audio data. A trade-off exists between the final audio quality and the compression techniques used in order to overcome the bandwidth requirements and achieve the real-time communication. So far, it has been pointed out that at this stage of development, Internet telephone systems are far from being a substitute for conventional telephones [3]. Until the current bandwidth increases significantly and a new real-time TCP/IP protocol emerges, a compromise in quality will remain.

3. PROPOSED APPROACH

As discussed in the previous section, there are three major problems which need to be tackled in order to build an Internet telephone system. These are the resolution of dynamic IP addressing; the adaptation of the Internet TCP/IP protocol to support real-time communication; and the provision of telephone conversation of compatible quality to a conventional telephone. We have adopted the following approaches in tackling these issues for the development of the Internet Telephone Software System.

A Telephone Exchange approach is used to resolve the dynamic IP addressing problem. This approach is widely adopted by most existing Internet telephone systems. In the telephone exchange approach, a dedicated server is used to maintain and manage information of active registered users running an identical set of Internet telephone software. Its main role is to facilitate finding and connecting users. Users using this approach must first register themselves with the telephone exchange by supplying some form of user identifier (ID) and IP address. The telephone exchange generally supports either a query-retrieval model or an Internet Relay Chat (IRC) [7] model for the retrieval of IP addresses. In the IRC model, users will log onto the server directly and use the navigation facility to browse and identify the intended recipient and IP address. Using the telephone exchange approach basically creates closed communities of users. The main attraction of this approach is the chat-line capability to allow users to find one another. From the developers’ point of view, it allows them to keep track of their customers base, monitor volume and usage patterns, and provide support and services (e.g. advertisers) for their customers.

Systems using this approach can use either a global server (e.g. FreeTel [8], CoolTalk [9], DigiPhone [10]) or distributed servers (e.g. Internet Phone 3.2 [11], Netphone [12]) to aid the resolution of IP addresses and connection process. In our
prototype, distributed telephone exchanges are used to maintain Internet telephone users. This allows the load to be spread out and eliminates the problems associated with an exchange going down. If this happens, users will simply need to register themselves with another telephone exchange. In addition, the system is designed to support direct connections given known IP addresses so that the connections are made through the host computers and not on the exchange itself.

The existing TCP/IP protocol is used in delivering the audio packets over the Internet. However, new mechanisms such as buffer management and packets lost replacement control are incorporated to make it possible to support real-time audio communication. This approach is aimed at delivering real-time communication at an acceptable quality of service. Currently, this approach is also used by all existing Internet telephone systems and until a new protocol emerges and becomes a standard, it appears that developers are contented to make the best use of current resources and limitations.

In using this approach, it is important to realise that proper sending and receiving of audio data though important is in itself insufficient. More importantly is the on-time delivery of data which is generally dependent on network performance. This in turn is largely related to the kind of network technologies implemented which the system has basically no control over. With many different network technologies available on the Internet, the system should not be made to rely on the underlying network infrastructure. Under such varying network load conditions, the audio packets will suffer varying degrees of delay. The variance in delay produces jitters which are undesirable for real-time services. In order to attenuate the jitters, the recipient’s host computer must provide some buffering mechanism. This will prevent the distracting utterance of audio, thereby permitting continuous playback. In addition, the unreliability of the network can give rise to packets lost and duplication which will deteriorate the voice quality. Hence, additional processing is required to make the system more resilient to these ill-effects of unstable network conditions to maintain a satisfactory overall performance.

In using the existing TCP/IP protocol, a choice can be made to use either TCP and UDP entirely or in some form of combination. Using the connection-oriented TCP alone ensures guaranteed delivery of control and audio data but will result in unacceptable levels of delay and jitters. Using the connectionless-oriented UDP with no guaranteed arrival of data is clearly not suitable for transmitting control data which is required for the functionality of the system. As such, the system utilises UDP for audio data transmission and TCP for control and other important data transmission.

In order to overcome the problems of jitters, non delivery and duplication of packets, incoming audio samples are packet time-stamped before they are transmitted to the recipient. In addition, redundant audio data pertaining to previous samples are combined with the current sample to make up an audio packet. This redundant data is used for the replacement of lost packets during the delivery process. At the recipient’s end, incoming audio data is placed in the correct position within the revolving circular buffer according to the time stamp. Lost packets are detected and corresponding made up and inserted in position. Continuous audio playback is achieved by continuously playing back the contents in the buffer.

Finally, the quality of audio communication delivered by Internet telephone systems is dependent upon the compression algorithms for encoding the audio data, processing power of the host computers and network transmission factors. These three factors are
inter-related and an optimal is sought to ensure satisfactory audio quality. For instance, using a high compression ratio in order to minimise network traffic will usually require high processing power and result in a potential drop in audio quality due to larger loss of original audio information. On the other hand, satisfying the processing power of the majority of the existing host computers will require the use of a less demanding compression algorithm but result in higher network traffic. This may lead to higher potential lost or delay of data packets which will have a detrimental effect on the quality of communication. It should be obvious that there is no clear cut solution to the problem. Likewise, there is no ideal compression algorithm which should be employed in such systems. The system must have some defined minimum level of processing power to carry out the various tasks, a fast enough modem to transmit and receive data to support the synchronous nature of the application and a number of compression algorithms to suit different network conditions. It would be advantageous if the system is self-adapting to changing conditions and optimally adjust itself to provide an optimal level of performance. Our prototype system supports a number of compression algorithms and allows audio characteristics to be changed to the desired level. The selection of compression algorithms and other parameters are set during a negotiation process prior to connection.

4. SYSTEM ARCHITECTURE

The system architecture of the Internet Telephone Software System is divided into two main components as shown in Figure 2. These are the Telephone Exchange and Telephone Software. The Telephone Exchange is basically a number of processes running on a dedicated server to manage a list of active users who are connected to the system. The Telephone Software component comprises an user interface for controlling the session and audio management processes. This segregation is necessary to cater for the different roles played by each process to support the real-time nature of the communication process.
4.1 Telephone Exchange

The Telephone Exchange plays the role of a telephone exchange to aid the connection process, monitor the quality of services and provide an estimation of characteristic parameters such as rate of lost packets, duplicated packets and delays. This information can be used to advice users of the intended connection by giving them an indication of the quality of conversation under current conditions. Call routing is provided in the form of returning the IP address of the recipient’s host computer in response to a caller’s request for connection. In order to perform this functionality, it maintains a database of all active users on its sub-net that are currently on-line. In addition to communicating with the database for information, the exchange also communicates with other identical Telephone Exchanges. This distributed form of architecture allows other sub-nets of users to be linked together. The Telephone Exchange comprises four processes: control, quality monitor, registration and network communication (TCP).

**Control Process.** The control process multiplexes incoming messages from the network to the quality monitor and registration processes. It differentiates the type of messages based on the content of the messages. However, all messages destined for the recipient flow through the control process without any processing. These messages are forwarded to the network communication (TCP) process for delivery. Another function of
the control process is to process a caller’s query to return the IP address of a recipient. If the recipient cannot be found on the current exchange, the control process will communicate with other remote Telephone Exchanges in an attempt to locate the recipient on other sub-nets. It will subsequently inform the caller on the outcome of the query request. At such, there is a need for the Telephone Exchange to keep a list of other remote exchanges.

**Quality Monitor Process.** The quality monitor process is responsible for determining the round trip delay from the local sub-net to other sub-nets which have an identical Telephone Exchange. It performs this function by periodically emitting a number of packets targeted on the remote exchange over which the round trip delay is to be measured. These packets contain the system clock value corresponding to the packet generation. The remote exchange on receiving these packets immediately resend them back to the originator. Based on the time difference between dispatch and arrival, the round trip delay can be computed. When all the packets have arrived, the average delay can be computed. This information is presented to the caller whenever a request for connection to a recipient is made. Although this information is not necessarily accurate, it does however provide some indication of the expected quality of communication. Based on this information, the caller can decide to go ahead with the communication or abort the session. Besides the round-trip delays, other factors such as packets lost rate and packets duplication rate are measured and reported as well. The actual round trip delay is estimated during the session management’s negotiation process between caller and recipient.

**Registration Process.** The process manages the registration and de-registration of active users which are connected to the telephone exchange. It also supports users’ request to return IP addresses of active users. The process builds up its database of user-id and host computer’s IP address as users activate their copy of the Telephone Software. When a user goes off-line and de-registers from the system, the corresponding entry is removed from the database. Besides communicating with local users, the registration process also supports request from other telephone exchange to return the IP address of any active user. The process maintains an up-to-date information of active users by periodically ‘pinging’ their IP addresses. Users failing the ping process will likewise be removed from the database. This is to cater for abnormal disconnection or termination of the software system which results in the failure to de-register from the telephone exchange.

**Network Communication (TCP) Process.** This last process is the network function needed to support communication among processes. It delivers without delay all messages generated by the upper layers of the control, registration and quality monitor processes. It establishes a local connection point (defined by the IP address and a logical port number) where all control messages are transmitted and received. After the messages have been sent, the connection is disconnected immediately. This is to allow message exchange with different destinations to take place without delays.
4.2 Telephone Software

4.2.1 Session Management

Session management is concerned with negotiating and monitoring of session parameters. TCP is used to transport control messages used for communication since it is not tolerable to any loss of messages which will compromise the functionality of the telephone system. Three processes are utilised in session management: control, negotiation and network communication process.

Control Process. The control process receives a caller’s request for connection with a recipient and attempts to establish a communication path between the two parties. Each user in the Internet telephone system is uniquely identified by an Internet call identification which is made up of a user ID and an associated IP address. This information is automatically registered with the telephone exchange upon system start-up. In order to make a connection, the control process will first contact its local Telephone Exchange and issue a request to retrieve the current IP address of the recipient’s host computer. A positive or negative respond is issued by the exchange depending on the availability of the recipient. With a positive reply and an associated IP address, the control process will subsequently initiate a call set-up with the other party. This is followed by the negotiation of audio parameters to be used in the communication process. In addition, the control process is responsible for call termination, which when activated, will initiate a connection tear-down procedure to notify the other party to leave the session.

Negotiation Process. The negotiation process handles the arbitration of communication parameters between both parties before actual communication can take place. During the negotiation process, the caller as the initiating party will propose a set of standard communication parameters to the recipient. The recipient will attempt to match and agree, failing which, both parties may modify and suggest new standards for the remaining unresolved parameters. With all communication parameters resolved, the negotiation process will signal the audio management process to initiate the conversation. However, if either party cannot agree on a common set of communication parameters and where a compromise cannot be reached, either party may reject the connection. In addition, the negotiation process plays another role to provide an estimate of the jitters which is likely to be experienced during conversation. This is done by inserting the system clock time into the negotiating packets prior to transmission and estimating the round trip delay when the packets return back to the originator. The round trip delay is the difference between transmitting and receiving time. Jitter is calculated from the difference between the maximum and minimum round trip delay. The amount of jitter experienced is passed to buffer management to allocate the necessary buffer space.

Network Communication (TCP) Process. This provides the transportation functionality to dispatch messages to its destination using TCP. It functions similarly to the Telephone Exchange’s network communication (TCP) process.

4.2.2 Audio Management

Audio management is responsible for audio transmission from the recording and playback operations to the transmitting and receiving operations. Audio transmission is periodic with audio sampled at regular time intervals and goes through a series of processes prior to being transmitted. Although, the occasional non-delivery of audio
packets will reduce the quality of communication, it will not be disastrous, provided that small audio samples are dispatched each time and that some in-built mechanisms exist to cater for lost data replacement. In this case, audio management uses UDP/IP to transport the audio packets. The following processes are required to fulfil the role played by audio management:

**Record/Playback (Audio).** The Record/Playback (Audio) process provides the necessary audio interface for recording, playing back and adjustment of audio characteristics. Users are allowed to modify play and record volume, play and record balance, monitor gain at play destination and record source. However, the attributes of number of channels, sampling rate and sample’s resolution are non-user modifiable but are set during the negotiation process. Samples of 20ms (corresponding to audio packet that will be transmitted to the receiver) of audio data are accumulated and passed down the pipeline for further processing. The corresponding playback process is performed by accepting audio samples from the previous processing unit and output it the audio device directly.

**Packets time-stamping.** Each packet of audio sample obtained from the external device is linearly time-stamped to indicate the instant of sampling. This allows the packets arriving at the recipient’s host computer to be ordered in the correct time sequence before being played back. As each audio packet contains 20ms of audio samples, the time-stamp mechanism uses 20 as the base value with increments of 20 for each new audio packet generated. Although, the transmission is carried out in sequential order, the packets may arrive out of order due to the different network paths traversed and varying network delays. The time-stamp is stored in the field preceding the audio data within each audio packet. It is recorded using a 32-bits field in each packet. With a value ranging from 0 to 4294967291, it will exhaust itself after 49 days before it is reset and restarted from 0 again. Both audio samples and the associated time-stamp are forwarded to the next stage to compress the packet size before transmission.

**Compression & Decompression.** The compression algorithms reduce the audio sample size in raw audio PCM format by encoding it in another format. Compression algorithms supported by the system include A-law, µ-law [13], adaptive differential pulse code modulation (ADPCM) [14], Groupe Speciale Mobile (GSM) [15] and linear predictive coding (LPC) [15]. The first four formats are able to maintain almost the same quality as the raw format. However, as LPC produces a lower audio quality at a lower information rate, it is used to compress the previous audio packet into ‘redundant’ audio information and bundled together with the current sample to form an audio data packet. This information is used for lost packet replacement if necessary (see subsequent description). Decompression performed at the recipient’s end reverts the compressed audio samples (both redundant and non-redundant audio) to its uncompressed format that is ready to be played.

**Buffer Management and Packets Ordering.** The purpose of buffer management is to cushion the out of order, late delivery and jitters experienced by the packets. It cooperates with packets ordering to arrange the audio packets in sequential order by using a ring buffer to store audio samples. It is organised into slots of sizes equivalent to the total uncompressed audio samples’ size of an audio packet as shown in Figure 3. Incoming packets are decompressed, identified by its associated time-stamp and placed accordingly into the correct location in the buffer. The total buffer space allocated is dependent on the amount of jitters experienced by the packets. Although the amount of jitters will vary
with time, it is a complex process to dynamically keep changing buffer size during an ongoing session to cater for these variations. As such, a margin of safety of 2 has been incorporated so that the total buffer space is set to be twice the number of packets that can fit within an average jitter period. With the amount of space allocated for buffering, the play-out point occurs when the ring buffer is half full. This margin of safety cushions the variations of delay exceeding the jitter period used in the calculation of buffer space.

![Diagram of Buffer Management and Loss Packet Replacement](image)

**Figure 3. Buffer Management and Loss Packet Replacement**

**Packets Lost Replacement Control.** This control is necessary to cater for any underlying unreliable network transmission which results in packets lost and delay exceeding the tolerable limit. Hardman’s [16] approach has been used to implement this control. In this approach, each voice segment is encoded into two packets so that in the event of a voice packet lost, a duplicated encoding in the following packet can be played back. In order to reduce overheads due to duplicate voice encoding, the first packet of a voice segment uses toll-grade compression, whereas the duplicated encoding of the same voice segment uses a simpler form of encoding to reduce the cost in both processing power and bandwidth. Hence, this implementation produces toll-grade audio quality inter-mix with sub-standard audio quality. This often redundant information is combined to form an audio packet. In the event that a packet that is scheduled to be played is not present in the buffer, it will replace this lost packet with the redundant audio information encoded in the next packet. However, if consecutive packets are lost, no redundant audio data can be used to replace the missing links. When this happens, this period of time will be replaced with silence at the host computer.

**Network Communication (UDP) Process.** This process utilises UDP to transmit audio data packets to its destination without any form of buffering. This mode of transmission does not ensure its arrival at the recipient’s end. It defines a local unreliable connection point (defined by the local IP address and a conceptual port number) for transmitting audio packets. Incoming packets are forwarded up to audio management for processing and playback on the recipient’s audio device.
5. COMMUNICATION PROCESS

Figure 4. Communication Process Between Two Parties using the Internet Telephone Software System

Figure 4 shows an operational example of the communication process of the prototype Internet Telephone Software System. In order to establish a voice communication between a caller (User A) and a recipient (User B), five phases are involved. These are Registration, Call Set-up, Audio Transmission, Call Termination and De-registration.

Users are first required to run the Internet Telephone Software System before voice communication can be established. During the Registration phase, the IP address of the user's machine is recorded by a Telephone Exchange. This enables the user to be identified by a static user-identity, so that other users can use it to initiate a communication link. After the registration phase, the user is ready to call or accept Internet telephone calls.
The Call Set-up phase begins with a caller initiates a telephone call to a recipient. The caller's Telephone Software issues a request to its local Telephone Exchange to locate the recipient machine’s IP address. If the recipient is registered to the same local Telephone Exchange and the request is successful, the recipient’s call identification is forwarded to the caller to initiate a connection. If the request is unsuccessful, the caller’s call identification (used to identify the originator of the request) and request is forwarded by the local Telephone Exchange to other remote Telephone Exchange who will search its database for a match. If a remote Telephone Exchange finds the corresponding entry in its database, it will return the IP address of the recipient’s machine to the local telephone exchange which will correspondingly convey the information to the caller. However, if either the remote Telephone Exchange cannot be contacted or no user with the call identification is found in the remote server, a negative response will be given to the local Telephone Exchange.

Upon receiving the recipient’s IP address from the Telephone Exchange, subsequent communication between the caller’s and recipient’s Telephone Software will bypass the exchange. In the connection establishment, the caller’s Telephone Software will try to create a connection channel with the recipient’s Telephone Software. The recipient may reject connection, or accept the connection and exchange a session number. All subsequent exchange of messages between both users will be identified by this session number.

After the connection is established, both users will start negotiating the audio parameters needed for audio transmission. These include the sampling rate, number of channels, number of bits per sample and type of compression used. Further negotiations are carried out if the proposed set of communication parameters is not agreeable to any one user. However, if either user cannot accept any of the communication parameters even after negotiation, either user may reject the connection.

During the Audio Exchange phase, the two users will exchange the digitised voice signals using the User Datagram Protocol (UDP) based on the agreed set of audio communication parameters.

The voice communication session ends when one of the users hangs-up. During this Call Termination phase, the terminating user’s Telephone Software will send a disconnection request to the other user. The other user will acknowledge it without further negotiation. The call termination only releases audio and buffer resources at both ends. The Telephone Software is still able to call or accept Internet telephone calls unless the user disconnects the Telephone Software as in the De-registration phase.

In the De-registration phase, the user disconnects from its local Telephone Exchange. This is accomplished with the user’s Telephone Software sending a message to its local Telephone Exchange for the proposed disconnection.

In addition to the Telephone Exchange’s ability to resolve users’ IP addresses and maintaining an up-to-date list of active users, it also periodically sends out “time” packets to other remote Telephone Exchanges to estimate the round-trip time. This is to give some indication of the expected audio communication quality between users who are registered with these pairs of Telephone Exchanges by providing information on the round-trip delay. In order attain better performance, users should register with a Telephone Exchange which resides in the same geographical network topology as the Telephone Software.
6. CONCLUSION AND FUTURE WORK

This paper has identified the key design issues for the development of an Internet telephone software system to support full-duplex real-time voice communication between two users who are connected to the Internet. These key design issues are considered and factored into the design and implementation of a working prototype system. This system is currently evaluated and will be used as a basis for further study and development.

As continuing work, a more structured approach to measure the quality of the system will be carried out. This can be used subsequently as a benchmark for comparisons with other systems. In this respect, a quality metric is being formulated to measure the quality of service (QoS) offered by the system. QoS can be measured along two dimensions: QoS of the network and QoS of the recording. A mixture of quantitative measurements and subjective assessment methods can be used for this purpose [17]. QoS of the network can be measured using the parameters of minimum and maximum delay giving the delay jitters, percentage of packets lost, percentage of the packets duplication and percentage of packets delivered late. This essentially a measure of the network condition which directly leads to the quality of audio generated at the recipient with respect to the source. QoS of recording pertains to the measurement of number of channels, sample resolution, sampling frequency and encoding mechanism. Conversation opinion tests carried under both laboratory conditions and field tests can be used to gauge users’ perception of the audio quality delivered by the system.

In addition, as a means to attain an improvement in quality of service, a dynamic self-adapting approach may be used to detect varying network conditions, computer CPU load and voice inputs to automatically adjust the various parameters of the system to deliver consistently good quality audio. For example, if the CPU cannot encode and decode the audio data fast enough for transmission and playback, or the network is too heavily loaded, then the sampling rate could be reduced and/or different compression algorithms utilised instead. The extent of such a problem can be gauged by measuring an audio drop rate which corresponds to the audio data which is received by the host computer but dropped due to insufficient CPU power to keep up with the decoding of data. Such an approach basically allows the system to dynamically adapt the audio transmission rate to network conditions as well as the host computers’ processing capability. It has been shown to be successful in delivering real-time video and audio in the World-Wide-Web using the Vosaic browser [18].

REFERENCES


[8] FreeTel Communications, Inc. *FreeTel 1.00* <URL: http://www.freetel.inter.net/>.


