An Approach to Real-Time Voice Communications over the Internet

S. Foo, K.V. Chin, and S.C. Hui
Division of Software Systems, School of Applied Science
Nanyang Technological University, Nanyang Avenue, Singapore 639798
E-mail: {asschui, assfoo}@ntuvax.ntu.ac.sg

ABSTRACT
This paper describes the approach taken for the development of an Internet telephone software system which functions like a conventional telephone to support real-time voice communications over the Internet. The system allows voice to be captured and digitised at the source machine. Using Internet as the transmission medium, these voice samples are then transmitted to the destination machine where they are received and played back in real-time on its audio device. The system has been implemented on Unix-based Sun Sparc workstations at the School of Applied Science, Nanyang Technological University.

1. Introduction

With declining cost of computer hardware, advances in computer technology and growing interest in Internet, commercial and shareware products have begun to emerge to provide communication services over the network. The growing enthusiasm also stems from huge potential cost savings by making it possible to make transcontinental phone calls at the prices of local telephone calls. Some of these research prototypes and commercial products, such as Internet Phone [1], CyberPhone [2], Netfone [3] etc., primarily offer real-time voice communications and other value added services. Although these products do support the basic functionality of communication, its quality is far been from satisfactory than those offered by telephone companies. It has been identified that the inferior quality of the communication is mainly due to the high transmission delay and heavy network load which can result in either delayed packet forwarding or discarding of packets. This paper describes the approach taken for the development of a telephone software system for real-time voice communications over the Internet. Section 2 discusses the basic architecture of the Internet telephone software and Section 3 describes the approach adopted in this work. Finally, conclusions are given in Section 4.

2. Internet Telephone Software

Constructing a telephone software that offers good performance (in terms of hearing perception) against the contradictory factor of low cost (in terms of CPU and network load) is an exciting
problem. Figure 1 shows the basic components of the Internet telephone software system. It clearly indicates that the quality of voice communications over inter-networks depends on the roles played by the terminating machines, network equipment and network protocols used.

![Figure 1. Internet Telephone Software](image)

The terminating machine can either be a workstation or personal computer with sufficient computation power and audio capabilities. Its responsibilities include acquiring/playback of audio through some audio input/output devices; compressing/decompressing of audio information for economic transfer; and receiving/transmitting of audio information. In general, it is often not economical to transfer raw Pulse Code Modulation (PCM) audio signals as this will severely burden the network to support yet another form of data (audio). One possible solution is to have efficient encoding algorithms of analogue voice to digital signals. Many such algorithms are already in existence. Some of these algorithms are documented in [4].

Under 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 receiving/transmitting function of the terminating machine must provide some buffering mechanism. This will prevent the distracting utterance of audio, thereby permitting continuous playback. Network performance will seriously affect the delivery time of data between two communicating machines. The performance is largely dependent on the network technologies implemented. With many different network technologies available on the Internet, the telephone software is usually not dependent on the kind of network implementation used.

All the networks in the Internet are inter-connected together using a common communication protocol, namely the TCP/IP protocol suite [5]. It 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 TCP/IP protocol applies well for non-temporal data transmission. However, the sending and receiving of audio data, though important, is in itself insufficient. More importantly is the on-time delivery of
information. TCP/IP does not have any provisions to support the real-time nature of the audio. Hence, additional information has to be supplied to accommodate the real-time nature of the service.

3. System Architecture

Figure 2 shows the system architecture of the Internet telephone software system. The architecture consists of two components: Telephone Software and Telephone Software Exchange Daemon. The daemon process manages users connected to the system. It acts as a “telephone exchange” where incoming calls are routed to its destination and monitors the quality of services such as the available bandwidth, rate of lost packets and delays. The Telephone Software component comprises of a user-interface controlling two processes: session management and audio management. This segregation is necessary to cater for the different nature of the two roles needed to support the whole session.

![System Architecture Diagram]

Figure 2. System Architecture

3.1 Session Management

Session management is concerned with negotiating and monitoring of session parameters. The control messages used for the communication requires reliable communication and is not tolerable
Control Process. The control process receives user request for communication with another party and tries to establish a communication path between the two parties. A person is uniquely identified by both a user-id and an IP address. It registers all the users (user-id) currently on-line and the associated IP address of the machine on which they are using. Hence, each caller and the called party have a unique set of user-id and IP address to identify themselves. The user-id and IP address is collectively known as the user’s call identification.

With the called party’s call identification, the control process will contact its local Telephone Software Exchange Daemon and request it to retrieve the actual IP address of the other party’s machine. Two situations occur here. First, the control process may receive a negative reply from the daemon, in which case, it will reflect unsuccessful connection to the user. In the second case, the control process receives a positive reply. Upon receipt of the requested IP address, it initiates a call set-up with the other party. This is followed by negotiation of audio parameters. The control process is also responsible for call termination. During this phase, the control process initiates connection tear-down procedures to notify the other party of the request 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 called party. The called party will attempt to match and agree, failing which, both parties may modify and suggest new standards for any of the communication parameters that is not agreeable to either party. After having agreed on the set of communication parameters, the negotiation process will signal the audio management process to start sending and receiving audio packets for the audio communication in accordance to the communication parameters agreed upon. 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 process also estimates the jitters that a packet may suffer by inserting the system clock time into the negotiating packets before the packets are transmitted. This allows estimation of the round trip delay when the packets return to the originator where it calculates the difference between transmitting and receiving time. Jitter is calculated from the difference between the maximum and the minimum round trip delay. The amount of jitter experienced is passed to the buffer management to allocate the necessary buffer space.

Network Communication (TCP) Process. The network communication process provides the transportation functionality to move the messages to the destination. It delivers without delay all messages generated by the upper layers of the control and negotiation 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.
3.2 Audio Management

Audio management is responsible for audio transmission from the recording and playback operations to the transmitting and receiving operations. The audio transmission is periodic with audio sampled at regular time interval and goes through the whole processing activities before being transmitted. It is time-sensitive in nature. 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. Therefore, audio management uses UDP / IP to transport the audio packets.

Record / Playback (Audio)  

Record / Playback (Audio) provides all the necessary audio interface for recording, playing back of audio and adjustment of audio characteristics. Users are allowed to modify play and record volume, play and record balance, monitor gain, and play destination and record source. However, the attributes: 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 an audio packet that will be transmitted to the receiver) of audio data are accumulated and pass down the pipeline for further processing. The corresponding playback process is performed by accepting audio samples from the previous processing unit and output to the audio device directly.

Packets time-stamping.  

Each packet of audio sample obtained from the external device is given a timestamp to indicate the instant of sampling. The purpose of time-stamping and the linear increment requirement is to order the packets in time sequence at the receiver end. As each audio packet contains 20ms of audio samples, the timestamp mechanism can, for simplicity, use 20 as the base value with increments of 20 each time a new audio packet is generated. Although, the transmission is in sequential order, the packets may arrive out of order due to the different network paths traversed and varying network delays. The timestamp is stored in the field preceding the audio data within each audio packet. Timestamp is recorded in 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 the packet of audio samples and the associated timestamp are forwarded to the next stage to reduce the packet size before transmission.

Compression & Decompression.  

The compression algorithms reduce the audio sample size (in raw audio PCM format) by representing it in another format. These include A-law, µ-law, adaptive delta pulse code modulation (ADPCM), GSM and code excited linear prediction (CELP). The first four formats are able to maintain almost the same quality as the raw format. However, as CELP has a lower information rate, it is used to compress the audio samples of the previous audio packet into ‘redundant’ audio information. Decompression performed at the receiving end reverts the compressed audio samples (both redundant and non-redundant audio) to the uncompressed format that is ready to be played.
**Buffer Management and Packets Ordering.** The sole purpose of buffer management is to cushion the out of order, late delivery and jitters experienced by the packets. It co-operates 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 audio samples’ size of an audio packet. As shown in Figure 3, received packets are decompressed and using the associated timestamp and accordingly placed into the corresponding location in the buffer.

![Buffer Management Diagram](image)

Based on the timestamp, 'received audio samples placed in sequential

- **Received Packets**
- **Decompression**
- **Primary audio data**
- **Redundant audio data**
- **Time-stamp**

Figure 3. Buffer Management

The total buffer space needed is dependent on the amount of jitters experienced by the packets. Although, the amount of jitters will vary with time, it is not possible to change the buffer size during an on-going session, as this will upset the arrangements. Hence, the total buffer space is set to the number of packets that can fit within two times a jitter period. With the amount of space allocated for buffering, the playout point is when the ring buffer is half full. The additional buffer space is used to accommodate any future variation of delay exceeding the jitter period used in the calculation of buffer space.

**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. In the event that a packet that is scheduled to be played is not present in the buffer, it will try to replace this lost packet with the redundant audio information encoded in the next packet. However, if consecutive packets are also 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 terminating machine.

**Network Communication (UDP/IP).** Network communication (UDP/IP) uses unreliable data delivery where packets are sent without ensuring its arrival at the receiver. It defines a local unreliable connection point (defined by the local IP address and a conceptual port number) for all incoming audio packets. Incoming packets are forwarded up the pipeline which will subsequently be output to the audio device. Packets due for transmission are sent without buffering.
3.3 Telephone Software Exchange Daemon

Telephone Software Exchange Daemon is a collective number of processes which performs mainly call routing and querying of quality of service parameters of the network. Call routing is provided in the form of returning the IP address of the machine a user is using in response to user’s request for communication. In order to perform this functionality, it maintains a database of all users that are currently on-line on its subnet. It stores both the user-id and the IP address of the user’s machine when the user starts up its copy of the telephone software. Besides call routing, it also estimates the network conditions which affect the conversational quality. The daemon comprises four processes, namely, 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 receiver flow through the control process without any processing. These messages are forwarded to the network communication (TCP) process for delivery. Another duty of the control process is representing local subnet’s users to query a remote daemon of the IP addresses of the other parties’ computers. When a local user wants to communicate with another remote user, the local Telephone Software passes a message to the local Telephone Software Exchange Daemon requesting it to retrieve the IP address of the remote user. The control process passes the request to the remote daemon on behalf of the local user. The result of the request will then be passed back to the user.

Quality Monitor Process. The quality monitor process is responsible for determining the round trip delay from the local subnet to other subnets which also have the Telephone Software Exchange Daemon running. It performs this function by periodically emitting a number of packets targeted on the daemon on a subnet over which the round trip delay is to be measured. The packets contain the system clock value corresponding to the packet generation. The remote daemon receiving these packets immediately resents them back to the originator. When the originator receives the packets, it will obtain the difference between the received and transmitted time of the packets. When all the packets have arrived, the average delay can be computed. The daemon provides this information to the user whenever a request to communicate with another party is made. Based on this information, the user will decide whether to go ahead with the communication or to abort the session.

Registration Process. The registration process performs the recording of all current on-line users and responds to all queries of its users computer’s IP address. The process builds up its database of user-id and the associated computer’s IP address as local subnet users activate their copy of Telephone Software. Apart from registration, the Telephone Software component also informs the registration process whenever it goes off-line. In this case, the registration process de-registers the user by removing its entry from the database. Besides communicating with local users, the registration process can also receive requests from remote daemons requesting for the IP address of any user.
Network communication (TCP) Process. The last process is the network function needed to support all the communication between the entities. As control messages are involved, reliable communication must take place so that TCP is used. This process functions similarly to the network communication (TCP) process of the session management process.

4. Conclusions

This paper has described the approach taken for the development of an Internet telephone software system which supports real-time voice communications over the Internet. It is currently implemented on a set of Unix-based Sun Sparc 5 workstations at the School of Applied Science, Nanyang Technological University. Performance evaluation of the system is being conducted to gauge the quality of service and users’ acceptance. The quality of service (QoS) serves as a metric for the quality of the connection. QoS can be measured along two dimensions: QoS of the network and QoS of the recording.

As seen from the design, the audio samples are generated in consistent intervals and undergo a compression process before being transmitted to the destination. In an ideal condition, the destination will receive all audio packets at constant time intervals in the order that they are transmitted and the audio playback will then have the same quality as generated at the source. However, in practice, this ideal condition is not present and the unreliability factors of Internet affects the delivery of the audio packets. Hence, QoS of the network is defined with the following parameters: minimum and maximum delay giving the delay jitters, percentage of packets lost, percentage of packets duplication and percentage of packets delivered late. This is essentially a measure of the network condition which directly leads to the quality of audio generated at the destination with respect to the source. Besides network characteristics, QoS also measures the quality of recording by measuring four other parameters. These are number of channels, sample resolution, sampling frequency and encoding mechanism. With this set of QoS parameters, the overall system performance is currently being measured according to a standard metric. These measurements will gauge the overall network performance to the user’s perceptual audio quality of the system.

References