Integrating Internet with PSTN networks for voice services

S.C. Hui, S. Foo and C.K. Yeo

School of Applied Science, Nanyang Technological University,
Nanyang Avenue, Singapore 639798
fax: (65) 792-6559
email: {asschui, assfoo, asckyeo}@ntuvax.ntu.ac.sg

Abstract

The potential of Internet telephony circumventing long distance rates is certainly appealing as one can talk to someone on the other side of the world for as long as desired at no more than the cost of the local Internet access and connectivity charges. This paper discusses an Internet Telephony Gateway that allows the delivery of voice services through the Internet to conventional telephones on the Public Switched Telephone Network (PSTN). The gateway comprises a hardware interface card, and two software components for session and audio management. The hardware interface interacts with the trunk line to obtain audio data, answer, initiate and terminate a call. Additionally, it converts audio signal into a digital form suitable for transmission over the Internet. The session and audio management interface with the Internet for connection set-up, monitoring of the quality of services, negotiation of session parameters and audio communication.

Keywords: Internet telephony system, Internet telephony gateway, real-time voice communication, voice service, PSTN.

1. INTRODUCTION

Internet telephony is a general term that applies to voice services delivered over the Internet. The potential of Internet telephony circumventing long distance rates is certainly appealing as one can talk to someone on the other side of the world for as long as desired at no more than the cost of the local Internet access and connectivity charges. Besides the benefits of cost savings, using the Internet as an alternative backbone for voice traffic makes it possible to integrate voice, data, graphics and text on the same network. This is certainly beyond the capability of conventional circuit switched telephony.
Currently, the most common Internet telephony application is the Internet phone [1] that provides real-time voice communication over the Internet. An Internet phone requires the use of computers at both caller and recipient ends. Another emerging Internet telephony application is the Internet telephony gateway that allows the delivery of voice services through the Internet to conventional telephones on the Public Switched Telephone Network (PSTN), without requiring the users to use a computer. One such system currently available is VocalTec’s Internet Phone Telephony Gateway Server [2] which is the earliest commercial product announced in 1996. Recently, another twenty firms have entered or announced intentions to provide Internet voice services for computer and/or conventional telephone users.

The objective of this research is to investigate the co-existence and interoperation of the Internet and PSTN networks. The focus is to integrate the Internet with the PSTN networks to allow users anywhere in the world to communicate with each other using a set of conventional telephones. To achieve this, an Internet Telephony Gateway is currently being developed at the School of Applied Science, Nanyang Technological University. The gateway comprises a hardware interface card and two software components for session and audio management. In this paper, we first introduce the concept of the Internet telephony gateway. Subsequently, the system architecture of Internet Telephony Gateway and its main components are then presented. The Application Programming Interfaces (APIs) for the gateway are also discussed. Finally, conclusions and future work are given.

2. INTERNET TELEPHONY GATEWAY

![Diagram of Internet telephony gateway](image_url)

Figure 1. Internet telephony gateway
The Internet telephony gateway essentially integrates the Internet with the PSTN networks as shown in Figure 1. In using the system, a caller will contact the local gateway server and request for a connection with a remote recipient. The local gateway will locate and contact the remote gateway using the standard Internet Transmission Control Protocol/Internet Protocol (TCP/IP) [3]. The remote gateway upon obtaining the connection request will contact the recipient. If this is successful, it will connect the two parties together and allow real-time voice conversation to take place with the Internet being used as the data transmission medium to channel the voice data back and forth to each other. Figure 2 shows a typical communication process of the gateway that is divided into four phases: Answer Call, Call Set-up, Audio Transmission and Call Termination.

**Figure 2. Communication process**

### 2.1 Answer call

A caller first calls the local telephone gateway by using a hunting directory number to access all the trunk lines to the gateway. The PSTN will send a ringing tone to the gateway through one of the free trunk lines. The gateway will detect the ringing tone and off-hook the trunk line.
2.2 Call set-up

With the trunk line being off-hooked, the gateway will request the caller to enter the area code of the recipient using an Interactive Voice Response (IVR) system of pre-recorded messages. If the gateway supports global telephone calls, the area code can be the standard IDD (International Direct Dialing) area code. Alternatively, if the server is used for a multinational corporation (MNC) with regionally or globally dispersed offices, the area code can be defined by a convenient user-defined extension number.

The gateway will make use of this area code to retrieve the IP address of the recipient’s gateway. Once the IP address is identified, the local gateway will attempt to establish a connection with the remote gateway using the Transmission Control Protocol (TCP) protocol. While waiting for gateway connection to be established, the caller is prompted by the local gateway to enter the recipient’s telephone number. This is to speed up the process as the two gateways may require some time for them to be connected. Data transmitted to the remote gateway include the trunk line ID, audio parameters, recipient’s telephone number and hang up control signal.

With the connection established, the remote gateway will check for available free trunk lines. A process ID is sent back to the local gateway so that all transactions between caller and recipient will be identified by this process ID. The remote gateway will off-hook the free trunk line and call the recipient. If the call cannot get through to the recipient, it will continue to try for a few times before informing the local gateway. All this while, the caller at the local gateway will be kept informed of the call status through the IVR system. Once the recipient is successfully connected to the remote gateway, the Audio Transmission process is activated.

2.3 Audio transmission

A set of audio parameters are first negotiated between the two gateways before audio communication can take place. These include the audio data sampling rate, number of channels, number of bits per sample, packet loss replacement technique and data compression technique. In order to speed up the negotiation, a series of preferred parameters are sent by the local gateway to the remote gateway. With the parameters negotiated and agreed, audio data can then be transmitted between the two gateways. Audio is first digitised, compressed and time stamped into packets before being sent to the other gateway. At the receiving gateway, the data packets are ordered, decompressed before being played back to the trunk line. Audio data is transmitted simultaneously in both directions between caller and recipient using the User Datagram Protocol (UDP) [3].

2.4 Call termination

When either party initiates a call termination by hanging up the telephone, the corresponding gateway will on-hook the trunk line and send a termination signal to the other gateway. The receiving gateway upon detection of this signal will correspondingly on-hook its trunk line. The Internet link between the two gateways is subsequently disconnected.
3. SYSTEM ARCHITECTURE

The Internet Telephony Gateway comprises a Hardware Interface Card, and two software components, namely, Session Management and Audio Management. The hardware interface card comprises a number of modules to convert the audio signal into a digital form suitable for transmission over the Internet; establish and terminate a telephone connection with the local trunk line; and a buffering mechanism to buffer and store encoded voice data. The hardware interface card is interfaced to the gateway computer that is connected to the Internet. Session Management and Audio Management interface with the Internet for connection set-up, monitoring of the quality of services, negotiation of session parameters and audio communication.

The two main transport protocols in the TCP/IP protocol suite, namely, TCP and UDP are both used to exchange information between gateways. TCP is a connection-oriented protocol with guaranteed delivery of data while UDP is a connectionless-oriented protocol with no guarantee of arrival of data. These two protocols apply well for non-temporal data transmission but does not have any provisions to support data transfer of real time nature. In order to support a real-time application using the TCP/IP protocol suite, extra mechanisms must be incorporated in the application layer. In order to achieve fast data transfer, no re-transmission of data is desirable thus making TCP unsuitable for data transmission. Therefore, UDP is used to transfer audio data and TCP is used to transfer control information and

Figure 3. System architecture
negotiation of communication parameters. The extra mechanisms that are required to support real-time transmission include data packet stamping prior to transmission, data buffering and data packet re-ordering upon receiving, and data packet loss replacement to cater for lost data packets during transmission.

4. HARDWARE INTERFACE CARD

![Block diagram of Hardware Interface Card](image)

The hardware interface card in the gateway computer supports the following system requirements to:

- detect incoming calls from the trunk line;
- answer incoming calls and prompt the caller for inputs via an IVR system using recorded voice messages;
- decode the DTMF tones generated from the telephone;
- convert the analog voice signals into digital form for storage and transmission between gateway servers, and vice versa; and
- initiate outgoing calls.

The hardware interfaces with the central processing unit (CPU) of the gateway computer which can be an IBM compatible PC with a EISA bus. Figure 4 shows the block diagram of the hardware interface card.
4.1 Line Interface Circuit (LIC)

The Line Interface Circuit provides a complete audio and signalling link between audio equipment and trunk line. Functions provided by the Line Interface Circuit include 2-4 wire conversion, loop seizure, external monitor telephone switch hook status and ringing voltage and loop current detection.

The 24 conversion circuit converts the balanced full duplex signal at Tip and Ring of the trunk line into a transmit ground referenced signal at VX (transmit). It also converts the receive ground referenced signal at VR (receive) into a balanced transmit signal at Tip and Ring of the trunk line.

The DC loop termination circuitry provides the loop with an active DC load termination when a logic low is applied to the LC (Loop Control) input. This is used to make the trunk line off-hook for both call answering as well as dial pulses generation.

The supervision circuitry is capable of detecting ringing voltage and loop current as well as the status of an optional external monitor telephone. The RVLC (Ring Voltage Loop Current Detect) output provides a logic low when loop current flows due to the external monitor telephone or due to the trunk line being in the off-hook mode.

4.2 CODEC

The CODEC (COmpression/DECompression technique) provides the conversion interface between the voiceband analog signals of the telephone subscriber loop and the digital encoded signals for transmission over the Internet. Analog (voiceband) signals in the transmit path entering VX, are sampled, quantized and encoded into digital form. The digital data output from DSTo is then stored in RAM 1 for transmission over the Internet. Analog signals in the receive path leave at VR after the reverse process is performed on the digital signals input at DSTi.

Different encoding techniques are available for the CODEC operation. The common codecs used in telephony systems include A-law and \( \mu \)-law Pulse Code Modulation (PCM) [4], Adaptive Differential Pulse Code Modulation (ADPCM) [5], Codebook Excited Linear Prediction (CELP) [6] and Global System for Mobile Communications (GSM) [7]. In this work, the A-law and \( \mu \)-law PCM codec are used as the preferred hardware chip due to its availability and low cost. Moreover, the reconstructed speech quality is excellent and its simple algorithm renders it useful for real-time performance.

4.3 Dual Tone Multiplexed Frequency (DTMF) Transceiver

The DTMF transceiver detects the DTMF tone generated by the telephone set (at IN) when numbers are pressed on the telephone keypad and decodes the number. The decoded digit is sent out via D0 to D3 to the CPU. It is also capable of generating the DTMF tone pairs at TONE based on the data inputs D0-D3 from the CPU. These generated DTMF tones are then sent via the trunk line to initiate a call.

4.4 Call Progress Decoder
The Call Progress Decoder provides information to permit the CPU to make decisions on whether to initiate, continue or terminate calls. A tri-state 3-bit output code indicates the presence of dial tone, audible ringback and busy signal. The Call Progress Decoder is signalled to start decoding by applying a signal at Clear_In and when the 3-bit output code is ready for reading, it will alert the microprocessor via Data_Valid.

4.5 Buffer (RAM)

Two Random Access Memory (RAM) banks are used. These are first-in-first-out (FIFO) RAM modules. They essentially act as buffers between the CODEC and the CPU as the CODEC and the CPU are performing input and output at different rates. RAM 1 is used in the audio encoding process to store digital voice data written by the CODEC. At the same time, it allows the CPU to burst-read stored data on a first-come-first-serve basis. Thus, the acquisition of voice data from the trunk line is independent of the retrieval of audio data by the CPU for transmission over the Internet. Conversely, RAM 2 acts as a buffer to allow the CPU to burst-write to it. At the same time, it allows the CODEC to read digital voice data for decoding into outgoing analog voice signal for the trunk line.

5. SESSION MANAGEMENT

Session management is responsible for connection process, monitoring the quality of services, providing an estimation of characteristic parameters such as rate of lost packets, duplicated packets and delay, and the negotiation of session parameters between the communicating gateways. 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 gateway. Four processes are utilised in session management: control, quality monitor, negotiation and network communication process.

5.1 Control process

The control process receives a caller’s input for the recipient’s area code. It will then convert this area code into the corresponding IP address of the remote gateway from a database that stores all the available IP addresses of gateways. Subsequently, the control process will initiate a call set-up and attempt to establish a communication path with the remote gateway. 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 remote gateway to leave the session.

5.2 Quality monitor process

The quality monitor process is responsible for determining the round trip delay from the local gateway to the remote gateway. It performs this function by periodically emitting a number of packets targeted on the remote gateway over which the round trip delay is to be measured. These packets contain the system clock value corresponding to the packet generation. The remote gateway on receiving these packets will immediately resend them back to the local
gateway. 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. In addition, the quality monitor process plays another role to provide an estimate of the jitters which is likely to be experienced during the audio communication. Jitters is calculated from the difference between the maximum and minimum round trip delay. The amount of jitters experienced is passed to buffer management to allocate the necessary buffer space. Besides the round-trip delays and jitters, other factors such as packets lost rate and packets duplication rate are measured and reported as well.

5.3 Negotiation process

The negotiation process handles the arbitration of communication parameters between both local and remote gateways before actual communication can take place. During the negotiation process, the local gateway as the initiating party will propose a set of standard communication parameters to the remote gateway. The remote gateway will attempt to match and agree, failing which, both gateways 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 audio communication. However, if either gateway cannot agree on a common set of communication parameters and where a compromise cannot be reached, either gateway may reject the connection.

5.4 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, quality monitor 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 gateways to take place without delays.

6. AUDIO MANAGEMENT

Audio management is responsible for audio transmission which 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.

6.1 Record/Playback (Audio)

The Record/Playback (Audio) process provides the necessary audio interface for recording/playing back of audio data from and to the truck lines via the hardware interface. The attributes of number of channels, sampling rate and sample’s resolution are set during the negotiation process. Samples of 20ms (corresponding to audio packet that will be transmitted to
the remote gateway) 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 remote gateway and output it to the truck line.

6.2 Packets time-stamping

Each packet of audio sample obtained from the truck line is linearly time-stamped to indicate the instant of sampling. This allows the packets arriving at the remote gateway 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-bit 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 to the remote gateway.

6.3 Compression and decompression

The (software) compression algorithms reduce the audio sample size in raw audio PCM format by encoding it in another format. Compression algorithms supported include A-law, μ-law, ADPCM, GSM and linear predictive coding (LPC) [8]. 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 remote gateway reverts the compressed audio samples (both redundant and non-redundant audio) to its uncompressed format that is ready to be played.

6.4 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 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 uncompressed audio samples’ size of an audio packet as shown in Figure 5. 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 on-going 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 jitters 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 jitters period used in the calculation of buffer space.


6.5 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 [9] 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 loss, 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 during playback.

6.6 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 remote gateway. 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.
7. Application programming interface

In order to support the functionality of the gateway according to the system architecture discussed above, two main groups of Application Programming Interfaces (APIs) are developed for the gateway which is running on a personal computer under Microsoft NT operating system. The first group of APIs is used to interface with the gateway hardware which interacts with the trunk line to obtain audio data, answer, initiate and terminate a call. Table 1 shows this group of APIs. The second group is to interface with the Internet for connection set-up and audio communication with remote gateways. As shown in Table 2, these APIs interface with the Internet to establish a connection between local and remote gateways, manage the data transmission, data reconstruction and playback, and optionally carry out data compression and decompression.

Table 1. APIs for gateway hardware interface

<table>
<thead>
<tr>
<th>API Functions</th>
<th>Hardware Modules Interfaced</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer_call</td>
<td>Line Interface Circuit</td>
<td>Send control signal to off-hook trunk line. API is activated by an interrupt from the hardware interface when it detects a ringing tone on the trunk line. An interrupt is preferred over polling of the status of the trunk line for reason of resource optimization.</td>
</tr>
<tr>
<td>decode_DTMF</td>
<td>DTMF Transceiver &amp; Line Interface Circuit</td>
<td>Decode DTMF tones from the telephone keypad. The output corresponds to the number pressed. This is used to extract the recipient’s telephone area code and number. Check which trunk line is available for the in-coming call to be connected. The API will return the identification number of the available trunk line.</td>
</tr>
<tr>
<td>check_trunk_line_</td>
<td>Line Interface Circuit</td>
<td>Monitor the status of the trunk line that include line free, ringing, dialing, talking and hang-up.</td>
</tr>
<tr>
<td>availability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>make_call</td>
<td>Call Progress Decoder, DTMF Transceiver &amp; Line Interface Circuit</td>
<td>Off-hook the available trunk line and send the telephone number through the trunk line using the DTMF transceiver on the hardware interface.</td>
</tr>
<tr>
<td>trunk_line_</td>
<td>Line Interface Circuit</td>
<td>Monitor the status of the trunk line that include line free, ringing, dialing, talking and hang-up.</td>
</tr>
<tr>
<td>monitoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td>record</td>
<td>Buffer (RAM 1)</td>
<td>Read audio data from the hardware interface. The size of the audio data read will depend on the audio sampling rate and the type of compression used.</td>
</tr>
<tr>
<td>playback</td>
<td>Buffer (RAM 2)</td>
<td>Reverse of the record API to send audio data received from the remote gateway to the hardware interface.</td>
</tr>
<tr>
<td>terminate_call</td>
<td>Call Progress Decoder &amp; Line Interface Circuit</td>
<td>Send control signal to on-hook (hang-up) trunk line. This API can also be activated by a remote signal to off-hook the trunk line when the remote trunk line has hung up.</td>
</tr>
<tr>
<td>IVR_playback</td>
<td>Line Interface Circuit</td>
<td>Playback pre-recorded audio to the trunk line for issuing instructions to the user.</td>
</tr>
</tbody>
</table>
Table 2. APIs for interfacing with the Internet

<table>
<thead>
<tr>
<th>API</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP_address_conversion</td>
<td>Convert the telephone area code into the corresponding IP address of the remote gateway. The IP address is retrieved from a database that stores all the available IP address of all the gateways.</td>
</tr>
<tr>
<td>establish_connection</td>
<td>Establish connection between two gateways.</td>
</tr>
<tr>
<td>send_available_trunk_line_ID</td>
<td>Send available trunk line identification number to remote gateway.</td>
</tr>
<tr>
<td>negotiation</td>
<td>Negotiate the communication parameters used for the sampling and transmission of audio data.</td>
</tr>
<tr>
<td>send_telephone_number</td>
<td>Send telephone number to remote gateway.</td>
</tr>
<tr>
<td>receive_telephone_number</td>
<td>Receive telephone number from remote gateway.</td>
</tr>
<tr>
<td>compression</td>
<td>Compress digital audio data to reduce size and speed up transmission.</td>
</tr>
<tr>
<td>time_stamping</td>
<td>Time stamp each audio data packet before transmission to remote gateway to enable incoming audio data packets to be ordered in the correct sequence for playback.</td>
</tr>
<tr>
<td>send_voice_datagram</td>
<td>Send compressed digital audio data packets to remote gateway.</td>
</tr>
<tr>
<td>receive_voice_datagram</td>
<td>Receive compressed digital audio data packets from remote gateway.</td>
</tr>
<tr>
<td>decompression</td>
<td>Reverse of compression API to decompress audio data. A common compression scheme must be utilized for both compression and decompression APIs.</td>
</tr>
<tr>
<td>packet_ordering</td>
<td>Re-arrange received audio data packets into the correct order before audio playback.</td>
</tr>
<tr>
<td>packet_loss_replacement</td>
<td>Replace audio data packets lost during transmission. Replacement schemes[6] supported include silence substitution, white noise substitution, waveform substitution, sample interpolation, embedded speech coding and redundancy techniques.</td>
</tr>
<tr>
<td>termination</td>
<td>Send termination signal to remote gateway to instruct gateway to hang up trunk line.</td>
</tr>
<tr>
<td>disconnect_link</td>
<td>Disconnect the network link between two gateways at the end of communication session.</td>
</tr>
</tbody>
</table>

8. CONCLUSIONS AND FUTURE WORK

This paper has discussed an Internet Telephony Gateway which integrates the Internet and PSTN networks to support real-time voice communication among users using conventional telephones connected to the PSTN. In addition to cost savings, such a system enhances the usability and versatility of the conventional telephone system and adds a new dimension to the telecommunication industry.

In addition to supporting conventional telephones, the gateway is being extended to bridge the connection between an Internet phone, Internet Telephone Software System [10] and conventional telephones on the PSTN networks so that these components can co-exist and inter-operate together. The objective is to achieve an integrated Internet service between Internet phones and conventional telephones. In doing so, Internet phone users can directly dial up conventional telephone users and carry out conversations. However, in order to achieve full interoperability among Internet telephony gateways and Internet phone products developed by different vendors, standards such as H.323 [11] should be adopted for real-time audio communication over the Internet. This issue is currently under investigation.
As further continuing work, it is recognised that in the current design, the session and audio management of the Internet Telephony Gateway employ the TCP/IP suite as the underlying transmission protocol over the Internet which does not guarantee any quality of services (QoS). Thus, it may be possible for the gateway to adopt a resource reservation set-up protocol such as the Resource ReSerVation Protocol (RSVP) [12] to guarantee the QoS over the Internet, thereby enhancing the audio quality delivered. However, the implementation of the RSVP protocol requires existing router software on the Internet to be modified or replaced. Thus, there will still be some way to go before RSVP can be really used for practical applications.

REFERENCES

3 Stevens, W. R. (1994) TCP/IP Illustrated, Volume 1, Addison-Wesley.