1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Design and Implementation of a PSTN/Internet Gateway for Internet Telephony

Cheng-Yue Chang and Ming-Syan Chen

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.
Email: {cychang@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw}

ABSTRACT

In recent years, the Internet has emerged as an important collaborative platform. Many applications utilize the Internet to provide new kinds of services. Among others, Internet telephony is attracting an increasing amount of attention for the reason that it has the potential to significantly reduce the cost of long distance voice communication. Moreover, Internet telephony gateways, or PSTN/Internet gateways, extend the cost saving benefits of Internet telephony to any one with a phone connected to the PSTN. In view of this trend, it is very important to construct a PSTN/Internet gateway for further experiments and developments. In this paper, we explore several methodologies for the implementation of a PSTN/Internet gateway. Based on these methodologies, we devise the corresponding architecture and implement a PSTN/Internet gateway. Following the ITU H.323 Recommendation, the gateway built is interoperable with other H.323 compatible terminals. We have validated the gateway built by having fluent two-way communication between a PSTN phone and a Microsoft NetMeeting terminal in our laboratory. A performance study on the sensitivity of the buffer size has also been conducted. It is noted that the gateway we devised is very versatile and can be easily extended to support various applications.

1. INTRODUCTION

Recent technology advances have brought revolutionary impacts to our life. The Internet emerges as a collaborative platform nowadays and it is envisioned to be even more important for years to come. Lots of applications utilize the Internet to provide new kinds of services, such as information services, entertainment, and communication services [16][18]. Search engines, for instance, provide information-searching services over the Internet. Distance education, as another example, makes it possible that the instructor and the students do not need to be physically present in the same location; it enables rural students to have the same opportunity for education as urban students.

Among others, real-time transmission of voice over the Internet, also known as Internet telephony, is attracting an increasing amount of attention. The reason for the importance of Internet telephony is that Internet telephony has the potential to significantly reduce the cost of long distance voice communication. Additionally, Internet telephony introduces entirely new and enhanced ways of communicating. Video conferencing, application sharing, and white-boarding are some typical applications that exploit real-time voice communication over the Internet.

PSTN/Internet gateways (Internet telephony gateways) extend the cost saving benefits of Internet telephony to any one with a phone connected to the Public Switched Telephone Network (PSTN). Consider the following scenario. One PSTN user in Taiwan wants to call another user in LA. He first connects to a local gateway in Taiwan, then routes via the Internet to a gateway in LA, and finally connects to the called party. The gateways are local to both calling and called parties. The gateway in the calling party side converts voice from the circuit-switched PSTN to packet-switched Internet. The gateway in the called party side works similarly. As the Internet is used for the transmission of long distance calls, expenses for these calls are then reduced. In addition to phone-to-phone communication, PSTN/Internet gateways can be used to provide phone-to-PC and PC-to-phone communication.

In view of the huge benefits the PSTN/Internet gateway can bring, it is of both theoretical and practical importance to construct one for further experiments and developments. Notice, however, that the PSTN is circuit-switched while the Internet is packet-switched. To interconnect these two intrinsically different wide-area-networks needs to take many aspects into account so as to overcome many technical challenges. Therefore, the goal of the paper is to design and implement a PSTN/Internet gateway for Internet telephony. Through this gateway, the PSTN user can connect to any Internet user who uses an H.323 [10][26] compatible terminal (such as Microsoft NetMeeting or Intel Internet Video Phone), and vice versa. In this paper, we explore several methodologies for the implementation of a PSTN/Internet gateway. Based on these methodologies, we devise the corresponding architecture and implement a PSTN/Internet gateway. As will be explained in detail later, the host application of our implementation can be further divided into three major components, i.e., PSTN Event Retriever, State Machine Manager and Audio Channel Manager. We have successfully implemented the PSTN/Internet gateway and validated the system built by having fluent two-way communication between a PSTN phone and a Microsoft NetMeeting terminal in our laboratory. A performance study on the sensitivity of the buffer size has also been conducted. It is noted that the gateway we devised is very versatile and can be easily extended to support various applications. For illustrative purpose, we extended our gateway to a call center providing voice-mail services over the Internet. The user may take advantage of this service to leave a message for the called party if the line is busy or there is no one answering the call. As a matter of fact, many other applications are conceivable. With the PSTN/Internet gateway prototype, we are able to do proper modifications to certain software components and obtain the customized applications we need very efficiently. This is the very advantage we have by implementing this PSTN/Internet gateway by ourselves.

The paper is organized as follows. We introduce the basic concept of Internet telephony, and the Internet telephony related standards in the Section 2. In Section 3, the architecture of the PSTN/Internet gateway that we propose is presented. We explain the implementation of the gateway in Section 4. Finally, we conclude this paper and describe the future works in Section 5.

2. PRELIMINARIES FOR INTERNET TELEPHONY

As described in Section 1, Internet telephony becomes an important application in our community. To facilitate our later discussion, we shall explore the basic concept of the Internet telephony and related standards in this section.

2.1 Introduction to Internet Telephony

In recent years, Internet telephony is even extended to the convergence of the telephone network (PSTN) and the data network (Internet) into a single communication network that offers powerful, economical, new communication options.

To understand Internet telephony, it is necessary to be familiar with the fundamental principles behind the Internet and how it compares to the PSTN. Although the Internet shares some characteristics with PSTN, it is very different from the latter. In Section 2.1.1, we will explore the differences between PSTN and the Internet. Section 2.1.2 will explain how these two different wide area networks converge as a single communication network.

2.1.1 The Internet vs. The PSTN

The PSTN is a circuit-switched network that has been optimized for real-time or synchronous voice communication with a guaranteed quality of service (QoS). When a telephone call is initiated, a circuit is established between the calling party and the called party. The PSTN guarantees the QoS by dedicating a full-duplex 64K circuit between the parties of a telephone conversation. Regardless of whether the parties are speaking or silent, they are using a 64K dedicated circuit until the call ends. Since the bandwidth remains constant, the cost of a telephone call on the PSTN is based on distance and time.

On the other hand, the Internet is a packet-switched network that has historically been used for applications where a variable QoS is tolerable, such as e-mail and file transfers. Packet switched networks do not dedicate a path between sender and receiver and therefore cannot guarantee QoS. It is noted that there have been some studies to provide QoS on the Internet [17][19].

All information to be transmitted over the Internet (voice, text, images, etc.) is divided into packets that contain a destination address and a sequence number. Internet routers and servers direct these packets over Internet until they arrive at their destination [24]. From the perspective of the Internet all packets are treated exactly the same regardless of their content. Once packets begin to arrive at their destination, the sequence number is used to put the packets into their original order. For applications where real-time interaction is not necessary, such as e-mail, the order in which the packets arrive or the delay between packets is not significant. However, real-time applications, like Internet telephony, can degrade considerably when there are long delays between packets. For this reason, packet switched networks have often been thought of as inappropriate for real-time applications like telephony. Real-time Internet communication is of importance. One scheme known as Realtime Transport Protocol (RTP) [20][21][22][23] proposed by IETF provides real-time end-to-end network transportation services. Another scheme known as Resource ReSerVation Protocol (RSVP) [15] is designed to reserve a certain amount of bandwidth specifically for Internet applications which require real-time response. It is believed that more and more schemes will be proposed to solve this problem.

2.1.2 Internet Telephony Gateway

Internet telephony gateway, or PSTN/Internet gateway, extends the Internet telephony to the convergence of the telephone network (PSTN) and the data network (Internet) into a single communication network. Internet telephony gateways take voice (or even a fax transmission) from the circuit-switched PSTN and place it on the packet-switched Internet and vice versa. Conceptually, an Internet telephony gateway works as follows. On one side, the gateway connects to the telephony world. It can communicate with any phone in the world. A phone line plugs into the gateway on this end. On the other side, the gateway connects to the Internet world. It can communicate with any computer in the Internet. The gateway takes the standard telephone signal, digitizes it (if it is not already digital), significantly compresses it, packetizes it for the IP, and routes it to a destination over the Internet. The gateway reverses the operation for packets coming in from the network and going out to the phone. The gateway also transfers and translates all the call control functions necessary for maintaining the call. Both operations (coming from and going to the phone network) take place at the same time, allowing a full-duplex (two-way) conversation.

Figure 1 shows a typical configuration, utilizing a single gateway for Phone-to-PC operation and two gateways for Phone-to-Phone operation. In the case of Phone-to-PC, the caller connects to an Internet telephony gateway over the PSTN. The Internet telephony gateway answers the call and prompts the caller to enter the IP address of the party he wishes to call. Then, the gateway will make a call to the called party. Once the connection is established, voice travels from the PSTN to the Internet between parties.

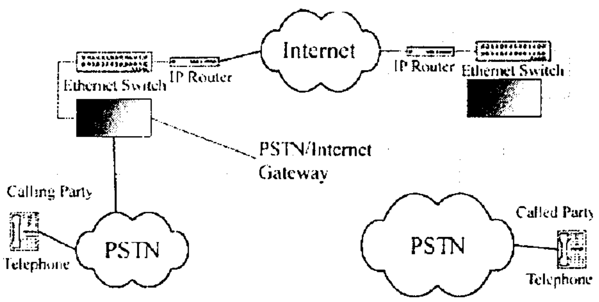

Fig. 1: Typical Internet Telephony Gateway Configuration

In the case of Phone-to-Phone, the caller also first connects to an Internet telephony gateway over the PSTN. The Internet telephony gateway answers the call and prompts the caller to enter the phone number of the party he wishes to call. The gateway will then look for another gateway that is local to the called party. The second gateway will attempt to locate the called party by placing a local PSTN call. Once the connection is established voice travels from the PSTN on and off the Internet between parties.

2.2 Overview of ITU H.323 Recommendation

H.323 is a recommendation from the International Telecommunications Union (ITU) that sets standards for multimedia communications over Packet Based Networks (PBNs). It was approved in 1996 by the ITU Study Group 15 [10][25]. The H.323 standard provides a foundation for audio, video, and data communications across IP based networks, including the Internet. By complying with H.323, multimedia products and applications from multiple vendors can interoperate, allowing users to communicate without concern for compatibility.

In essence, H.323 is an umbrella recommendation. It includes audio/video codecs (G.711 [1], G.722 [2], G.723 [3], G.728 [4], G.729 [5], H.261 [8] and H.263 [9]), call control (H.245 [7] and Q.931 [11]), PBN interface (H.225.0 [6]), and data conferencing (T.120 [12]) recommendations. It covers the technical requirements for audio and video communication services in PBN that do not provide a guaranteed quality of service (QoS). Figure 3 describes the scope and the major components of a H.323 system. Note that the scope of H.323, shown in Figure 2, does not include the PBN itself or the transport layer.

The major H.323 components are Terminal, Gatekeeper, Gateway and MCU (Multipoint Control Unit). Since our work is to design and implement a PSTN/Internet gateway, we will focus on the H.323 Gateway, and the description of the characteristics of these components will be omitted.

As in Figure 2, H.323 Gateways interconnect the H.323 system and other ITU-compliant systems (including PSTN), and provide translation functions between H.323 conferencing endpoints and other ITU-compliant terminals. The translation functions involve translation not only between transmission formats, communication procedures, but also between audio and video codecs. It performs call setup and clearing on both the PBN side and the circuit-switched network side. We have included all of these mandatory translation functions in the PSTN/Internet gateway we implemented.

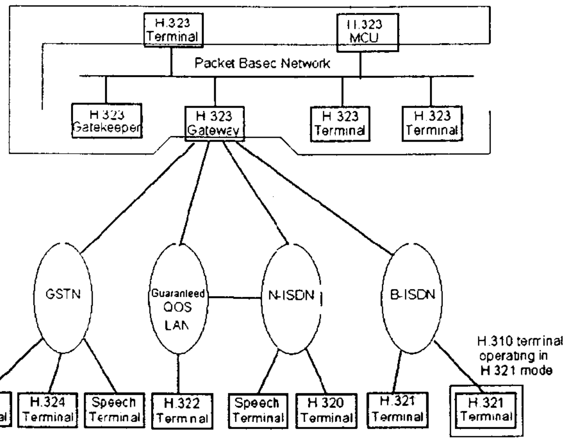

Fig. 2: The scope of H.323 (copy from H.323)

3. METHODOLOGY FOR BUILDING AN PSTN/INTERNET GATEWAY

In this section, we will devise methodologies to implement our PSTN/Internet gateway. Section 3.1 explains how to employ a Dialogic D/41ESC card as the interface to the PSTN for our gateway, and describes how we utilize an H.323 Protocol for our work. We design state machines for the gateway to maintain existing calls in Section 3.2. Architecture of our gateway is described in Section 3.3.

3.1 Adopting Dialogic D/41ESC PSTN Interface Card

Since the PSTN/Internet gateway interconnects the PSTN and the Internet, it consists of the corresponding interface to deal with these two different networks. On the Internet side, we utilize a typical Ethernet interface card to provide the access functions to the Internet. On the other hand, we utilize a Dialogic D/41ESC PSTN interface card to help us deal with the signaling of the PSTN, A/D or D/A conversion, and voice coding (A-law or μ-law).

The Dialogic D/41ESC PSTN interface card has four independent voice processing ports, and can offer four PSTN telephone lines. Each line is, however, half-duplex (one-way) processing, meaning that recording and playing of sound over one telephone line cannot work concurrently. To remedy this, we use two ports to perform full-duplex (two-way) communication so that our PSTN/Internet gateway can offer two calls with the Dialogic D/41ESC PSTN interface card. It is noted that the effect of echo must be considered in this situation. Our gateway makes use of the driver of the Dialogic PSTN interface card to deal with this problem. Thus, our gateway can provide echo-cancelled two-way communication.

1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Accompanying the PSTN interface card, Dialogic API [13] provides programmers with an application program interface to access the card. Dialogic API includes device management functions, I/O functions, call status transition event functions and speed/volume functions. Among these functions, call status transition event functions are most useful for us. We use them to detect the events from the PSTN. For example, as will be seen in Section 3.3, by calling these functions we can detect if a call from the PSTN occurs.

As the goal of our work is to build an H.323 compatible PSTN/Internet gateway, we shall have a protocol stack that implements the H.323 protocol. To this end, we utilize an H.323 Protocol Stack in our gateway. The H.323 Protocol Stack implements the mandatory components of H.323 protocol as function libraries. We make use of the H.323 protocol stack by calling these functions and receiving stack callbacks that notify us of the state transition in the H.323 Protocol Stack. Similarly to the call status transition event functions of Dialogic API, it is useful for us to receive stack callbacks. With these callbacks we can detect the event from the H.323 protocol stack. For example, we can detect that an H.323 call is disconnected by the remote endpoint.

3.2 Designing State Machines

A call is a point-to-point multimedia communication between two endpoints. The call begins with the call setup procedure and ends with the call termination procedure. Since each call has its own progression, for example, Idle, Offering, Connected, Disconnected, etc, we can utilize a state machine to manage the call. The key elements of a state machine are initial state, final state, intermediate states and transitions between states. We will define the call-states, call transitions and state diagram for our PSTN/Internet gateway in this section.

3.2.1 Call States

A call-state is clearly defined as a stage of a call progression. A call is created in one of the following two conditions:

- The H.323 Protocol Stack notifies that a call from the Internet is presented.
- The PSTN interface card notifies that a call from the PSTN is presented.

For a newly created call, the initial state is set to Idle. Then, each call moves from state to state as the call connection is established, connected, and concluded. The particular state of a call is in determining what commands and actions can be performed. Such states are called valid states for a particular command. If a command is issued when the call is not a valid state for that command, an error occurs.

3.2.2 Call-State Transitions

Each call maintains a current state. The transition from the current state to a new state is triggered by messages or events relating to the call. For instance, a call that is currently in the Wait_For_Disconnect state will transit to the Idle state if the remote party disconnects.

However, how can we know if any event happens? We can divide this question into two sub-questions. The first one is how the events from the PSTN side are detected. Based on the Dialogic API described in Section 3.1, we implement a module, PSTN event retriever, to detect the occurrence of the events. The Dialogic PSTN interface card adds a message to its own event queue while its status is changed. For example, it adds DE_RINGS to the queue if it discovers an incoming call is presented, or it adds TDX_PLAY to the queue if it completes the playback of sound over one port. Hence, we can detect the events from the PSTN by using the call status transition event functions of Dialogic API to keep retrieving events from the event queue.

The second sub-question is how the events from the Internet side are detected. To detect the events from the Internet side or H.323 system side, we first set up the event handlers by calling the library functions of the H.323 Protocol Stack, and then receive the stack callbacks.

3.2.3 State Diagram

Figure 3 shows a constructing element of a state diagram. Each state is represented by an ellipse that contains the state name. The states are connected with arrows indicating the valid call-state transitions. Each call-state transition includes two parts. The upper one is the message used to cause that state changed and the initiator of the message. The lower one is the responding actions of the host application. According to the principle we can construct the state machines for a call form the PSTN to the Internet and a call from the Internet to the PSTN.

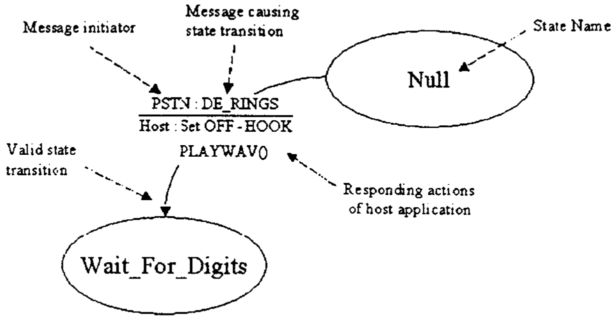

Fig. 3: State diagram explanation

3.3 Architecture of Our Internet Telephony Gateway

Based on the methodologies proposed in the previous sections, we construct architecture for our PSTN/Internet gateway. The gateway consists of four layers and is depicted in Figure 4. The four layers are composed of a network interface layer, an operating system layer, an application program interface (API) layer and a host application layer.

1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

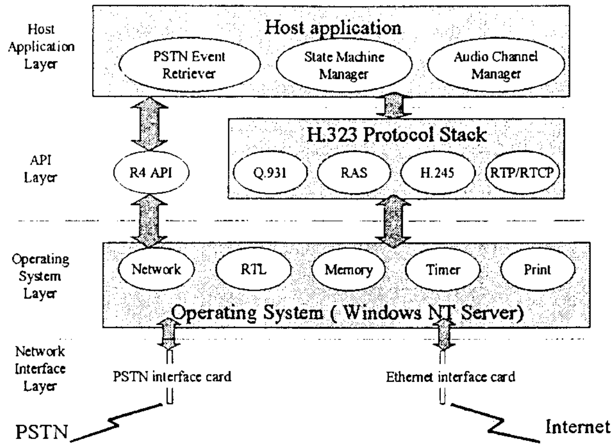

Fig. 4: The architecture of our Internet telephony gateway

The network interface layer is the bottom of the architecture, and is the physical layer of the architecture. It consists of two network interface cards, Ethernet interface card and PSTN interface card.

The operating system layer manages the network interface layer, and provides the upper layer with the common operating system services. As depicted in Figure 4, these services include network operations, dynamic memory allocations, run-time libraries, system timer, and print. We adopt Microsoft Windows NT Server as our operating system because of the interoperability between API and the NT operating system layer and also the ability of the NT operating system to execute programs in the multi-thread mode.

The application program interface layer consists of Dialogic API and the H.323 Protocol Stack API. As mentioned in the previous sections, Dialogic API provides programmers with a set of libraries to make use of the PSTN interface card while H.323 Protocol Stack API implements the mandatory elements of H.323 as function libraries.

The host application consists of three major modules: State Machine Manager (SMM), PSTN Event Retriever (PER) and Audio Channel Manager (ACM). The function blocks of the host program are shown in Figure 5. For existing calls, the host application maintains independent state diagrams in the system. Hence, we design a module, State Machine Manager, to manage these state diagrams. The SMM keeps track of the call-state for each call, and triggers the call-state transitions by retrieving events from the PSTN interface card and the callback from the H.323 Protocol Stack. The responsibility of the PSTN Event Retriever is hence to keep retrieving events from the event queue of the PSTN interface card.

Audio Channel Manager (ACM) is designed to conduct the buffering between PSTN interface card and the H.323 protocol stack. ACM is created for each call after the audio channels have been opened. For each call from PSTN to Internet (explicitly, NetMeeting in this very case), it uses the I/O functions of Dialogic API to retrieve the audio data from the PSTN interface card and buffers it with our buffering mechanism. Then it packetizes and transmits the audio data to the called party by the library functions of the H.323 Protocol Stack. For the data flow toward the other direction, ACM receives and unpacketizes the audio packet by the library functions of the H.323 Protocol Stack. Then, it buffers the audio data, and finally uses the I/O functions of Dialogic API to playback it.

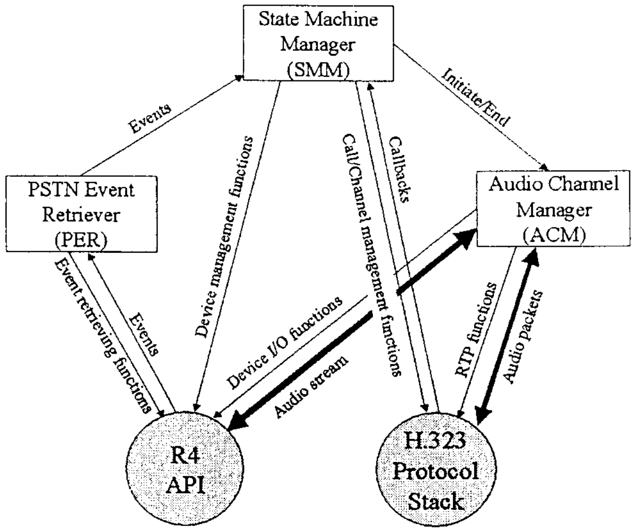

Fig. 5: The function blocks of host program we devised

4. IMPLEMENTATION OF THE PSTN/INTERNET GATEWAY

In this section, we describe the implementation of the PSTN/Internet gateway. Section 4.1 describes the implementation of the PSTN Event Retriever. Section 4.2 explains how we implement the Audio Channel Manager. The implementation of State Machine Manger is presented in Section 4.3. Section 4.4 illustrates an example to show that the gateway we devised is very versatile and can be easily extended to support various applications. A performance study on the sensitivity of the buffer size has also been conducted in Section 4.5.

Our PSTN/Internet gateway is developed on IBM compatible PC with Pentium II 266 CPU and 128 Megabytes RAM. The operating system of our gateway is Windows NT Server Version 4.0. The network interface to the Internet is 10-base-T Ethernet card while the network interface to the PSTN is Dialogic D/41ESC PSTN interface card. We use Microsoft Visual C++ 5.0 as the compile tool for executable programs.

4.1 Implementation of PSTN Event Retriever

As described in Section 3.3, the host application uses the module, PSTN Event Retriever, to detect the events from the PSTN interface card. PSTN Event Retriever is implemented as a new working thread created by the host application. The work of this thread is keeping detecting the events from the PSTN interface card and reporting to the host application. Thus, we utilize an endless loop (while(1)) within the thread to keep retrieving events. In the loop, the PSTN Event Retriever retrieves the events by calling the following functions of Dialogic API:

- sr_getevtdev() — gets the initiative device (port) for the current event.
- sr_getevttype() — gets event type for the current event.

As the program ends, this thread is also killed.

4.2 Implementation of Audio Channel Manager

As mentioned in Section 3.3, Audio Channel Manager is used to buffer audio data between PSTN interface card and H.323 Protocol Stack. Upon the establishment of the audio channels, the host application creates a new working thread for the ACM module. The work of the ACM module is as follows. For a call from the PSTN to the Internet, the ACM first records the voice from PSTN interface to the buffer. Then it sets the system timer to periodically call the function "sentrtp" to packetize and send the audio packets.

For a call from the Internet to the PSTN, audio packets are received from the H.323 protocol stack by the H.323 callback function "rtpreceive". "rtpreceive" unpacks and puts the audio data to the buffer, and then ACM plays the audio data from buffer to the PSTN interface card.

4.3 Implementation of State Machine Manager

After our gateway has been initiated, the responsibilities of the State Machine Manager are to maintain the current state of each state diagram, and to wait for events to trigger the corresponding call-state transitions.

We create four text windows to keep track of the status of each device (channel) which is reported by the PSTN Event Retriever. As shown in Figure 6, the four devices are opened and set to ON_HOOK after the initialization procedure.

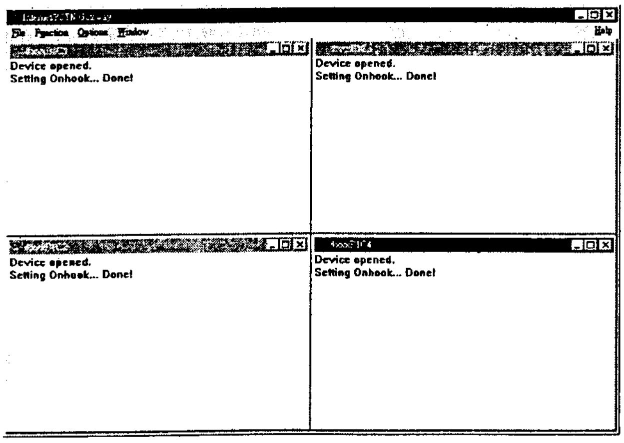

Fig. 6: GUI to display the status of the four channels on the PSTN card

For the event from the H.323 protocol stack, we create a dialog to display the status of the protocol stack. As shown in Figure 7, the left part of the dialog shows the progress and the information during Q.931 call setup procedure while the right part of the dialog shows the events from the H.323 protocol stack. After the initialization procedure, the right part shows that our gateway is listening to the socket port number 1720, and also the telephone numbers of our gateway.

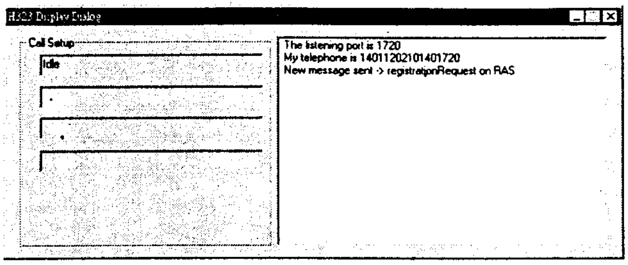

Fig. 7: GUI to display the status of H.323 Protocol Stack

According to the event received by the State Machine Manager, it triggers the call-state transition of the corresponding state diagram. For calls of two directions (from the PSTN to the Internet and also vice versa), we design two kinds of state diagrams to specify the valid call-states and call-state transitions.

4.4 Developing Customized Application

It is worth mentioning that the gateway we devised is very versatile and can be easily extended to support various applications. For illustrative purposes, we extended our gateway to a call center providing voice-mail services over the Internet. Consider the following example, a PSTN user wants to make a call to his friend on the Internet. However, the line is busy or there is no one answering the call. In this case he would like to leave a message for his friend. Our call center is able to support this very service. The call center asks the caller to leave a message whenever he fails to make a call. If the user would like to leave the message, the call center asks for the receiver's E-mail address and begins to record his message to a WAVE file. Then it encodes the file according to MIME [14] format and sends the file as an email to the receiver.

To provide this service, we insert two new states, Wait_For_Address and Wait_For_Message to the state machine for a call form PSTN to the Internet. The host application is notified that one call is disconnected, and then it prompts the PSTN user to enter the receiver's E-mail address (i.e., by calling dx_play() and dx_getdig()). The state then transits to Wait_For_Address. Once the user has finished entering the receiver's E-mail address, the host application is notified by the message, TDX_GETDIG, retrieved by PSTN Event Retriever. It then begins to record the message (i.e., by calling dx_record()), and the state transits to Wait_For_Message.

Wait_For_Message waits for the completion of recording the message until the host application is notified by the message, TDX_RECORD, retrieved by PSTN Event Retriever. It then saves the recorded file for later encoding and sending. The state transits to Idle. Note that within the above two states, the PSTN user may hang up the call, and the state transits to Idle directly. As a matter of fact, many other applications are conceivable. With the PSTN/Internet gateway prototype, we are able to do proper modifications to certain software and obtain the customized applications we need very efficiently. This is the very advantage we have by implementing this PSTN/Internet gateway by ourselves.

4.5 Performance Analysis

The performance of our PSTN/Internet gateway is defined as the continuity of the speech from the remote party. The primary effective variable of the continuity is related to inter-arrival jitters between received audio packets. In order to improve the performance of our gateway, we should minimize the number of the occurrences of inter-arrival jitters. The approach we employed is to pre-buffer some audio packets in the Audio channel Manager. As a result, when an inter-arrival jitter is long, Audio Channel Manager can still consume the pre-buffered data to keep the playback speed over the active voice channel on the PSTN interface card. The more data pre-buffered, the less likelihood the system buffer will become empty. However, having the more data pre-buffered means the longer delay between the speaking party and the receiving party. Therefore, we have to make a trade-off between pre-buffered data size and the times which the buffer is consumed to be empty.

To choose an optimized pre-buffered data size for our gateway, we measure the times when the buffer becomes empty under various pre-buffered data sizes. Two experiments, of 3 minutes and 5 minutes, are conducted.

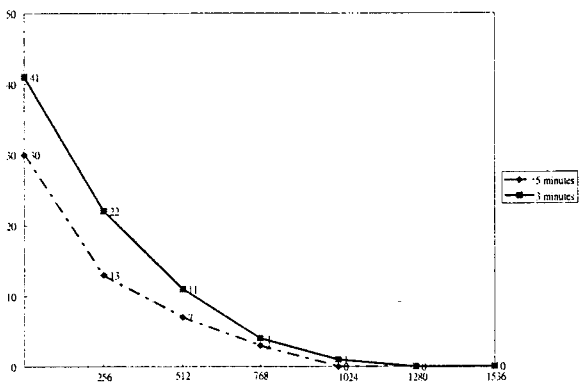

Fig. 8: Experimental Result

As shown in Figure 8, the experimental result indicates that when the pre-buffered data size is set to be 1024 bytes, the times when the buffer becomes empty is almost 0, implying that a pre-buffered data size of 1024 bytes is proper for this experiment.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we designed and implemented a PSTN/Internet gateway for Internet telephony. The gateway is designed to follow the ITU H.323 Recommendation, and is hence interoperable with other H.323 compatible endpoints on the Internet. By this gateway, we not only interconnect the PSTN and the Internet, but also converge the advantages of the two wide-area-networks.

We proposed architecture for our PSTN/Internet gateway in this paper, and based on the architecture we implemented the PSTN/Internet gateway with Microsoft Visual C++, and conducted some experiments in our laboratory. Moreover, we have identified several key issues which could lead to further improvement.

Our gateway currently only supports the mandatory audio codec, ITU G.711 (A-law or μ-law). The other codecs recommended by ITU H.323 include G.722 [2], G.723 [3], G.728 [4] and G.729 [5]. We can extend our gateway to support these optional codecs in the future. In addition, conventional telephones can only transmit audio while video phones can transmit both audio and video. It is noted that the price of the video phone is currently too high for the public consumers to afford it. However, it will become popular as long as the price of the video phone drops. Therefore, we can extend the capability of our gateway to support the video in accordance with this trend.

Furthermore, one configuration of H.323 gateway is that the gateway has the characteristics of H.323 Multipoint Controller. It involves the mixer design and the multicast technology of the Internet Protocol. The multicast is an important technology for H.323 conference call, and is thus a matter of our future research.

ACKNOWLEDGEMENT

This work was supported in part by the Computer Communication Laboratory of Industrial Technology Research Institute, Project No.G4-87029-D, Taiwan, Republic of China.

6. REFERENCES

[1] CCITT/ITU-T Recommendation G.711, "Pulse Code Modulation of Voice Frequencies"

[2] ITU-T Recommendation G.722, "7 KHz Audio-coding within 64Kbps"

[3] ITU-T Recommendation G.723, "Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 Kbps Using Low-delay Code Excited Linear Prediction"

[4] ITU-T Recommendation G.728, "Coding of Speech at 16 Kbps Using Low-delay Code Excited Linear Prediction"

[5] ITU-T Recommendation G.729, "Coding of Speech at 8 Kbps Using Conjugate-structure Algebraic Code Excited Linear Prediction"

[6] ITU-T Recommendation H.225.0, "Media Stream Packetization and Synchronization on Non-guaranteed Quality of Service LAN's"

[7] ITU-T Recommendation H.245, "Control Protocol for Multimedia Conferencing"

[8] ITU-T Recommendation H.261, "Video Codec for Audiovisual Services at p x 64 Kbps"

[9] ITU-T Recommendation H.263, "Video Coding for Low Bit Rate communication"

[10] ITU-T Recommendation H.323, "Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-guaranteed Quality of Services"
[11] ITU-T Recommendation Q.931, "ISDN User-network Interface Layer 3 Specification for Basic Call Control"
[12] ITU-T Recommendation T.120, "Data Protocols for Multimedia Conferencing"
[13] User's Manual, "Voice Programmer's Guide for Windows NT", Dialogic Corporation, Aug 1996
[14] N. Borenstein, N. Freed, "RFC 1521: MIME (Multipurpose Internet Mail Extensions) Part 1 – Mechanisms for Specifying and Describing the Format of Internet Message Bodies", September 1993
[15] R. Braden Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", September 1997
[16] B. Braden, D. Clark, S. Shenker: Integrated Services in the Internet Architecture: an Overview, RFC-1633, June 1994.
[17] A. Campbell, G. Coulson and D. Hutchison, "Supporting Adaptive Flows in Quality of Service Architecture", ACM Multimedia Systems Journal, 1996.
[18] F. Fluckiger: Understanding Network Multimedia Applications and Technology.
[19] E. W. Knightly and P. Rossaro, "Improving QoS through Traffic Smoothing", Proceedings of IFIP IWQoS'96, Paris, France, 1996.
[20] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RFC 1889: RTP – A Transport Protocol for Real-Time Applications".
[21] H. Schulzrinne: The Real-Time Transport Protocol. MCNC 2nd Packet Video Workshop, Dec 1992.
[22] H. Schulzrinne: RFC 1890 – RTP Profile for Audio and Video Conferences with Minimal Control
[23] H. Schulzrinne: Some Frequently Asked Questions about RTP. http://www.cs.columbia.edu/~hgs/faq.html
[24] W. R. Stevens, "TCP/IP Illustrated, Vol. 1: The Protocols", Addison-Wesley Professional Computing Series, 1994
[25] G. A. Thom, "H.323: the Multimedia Communications Standard for Local Area Networks, "IEEE Communications Magazine, pp. 52-56, Dec. 1996.