STUDY AND APPLY SIP PROTOCOL TO IMPLEMENT A VOIP SYSTEM FOR A MEDIUM-SIZE ENTERPRISE NETWORK

PHẠM HỒNG LIÊN, TRƯƠNG TẤN ĐỨC ANH, BÙI QUANG HUY

Division of Telecommunication, department of Electrical-Electronic
HoChiMinh city University of Technology
Email: phlien@yahoo.com, ducanhtruongtan@yahoo.com, buiquanghuy2002@yahoo.com

ABSTRACT

This article introduces the basic issues of SIP, one of point-point call signaling protocols in the range of protocols for VoIP, from concept, position and role of SIP in a VoIP system to the elements participating in a SIP call as well as the main operation of these elements. This article also analyzes the core of this protocol like dialog and transaction as well as message's structure. After study the protocol in details, this article will present a design of an application system for transmitting voice by PCs in networks of small and medium-size enterprises. With the program written by Visual C++ 6.0 including main functionalities of SIP, and the processing of voice data in real-time, this system is easy to implement and is highly applicable as well as extendable.

Keyworks: SIP, VoIP system using SIP, SIP_UAC, SIP_Proxy, SIP events.

1. INTRODUCTION TO SIP

1.1. History and definition of SIP

SIP (Session Initiation Protocol) is a protocol used to initiate a session. As other point-to-point signaling protocols, SIP is deployed with the main purposes of initiation, modification and termination multimedia sessions.

SIP protocol occured at the end of decade 90' and is taken the initiative by prof Henning Shulzrinne and his research team in the department of Computer Science, University of Columbia, USA. In 1999, first specification of SIP, RFC 2543 [9], appeared. While disputation about the feasibility of SIP is heated, 2 years later, it is completely filled up with RFC 3261 [8]. The appearance of this specification made an obvious advance in the development of this protocol. From now on, SIP is on the right place in VoIP system. After that, a succession of specifications related to SIP occurs continuously.

1.2. Position and role of SIP

There are two important basics in the Internet phone application, process of call-control and process of voice transmission in the packet form. Nowadays, many open protocols are promoted to resolve these both basics, gathering to VoIP stack.

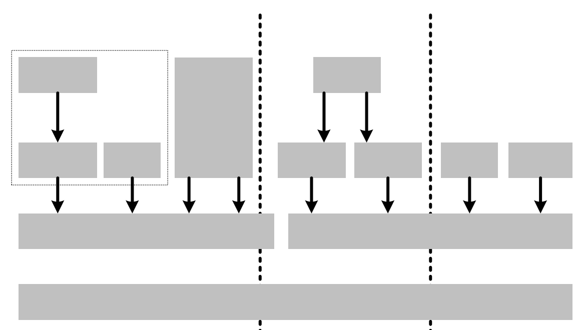

Figure 1: SIP in VoIP stack

The most popular protocols for call-control may be devided into two main groups:

- *Group of controlling multimedia gateway:* includes SGCP, IDCP, MGCP, Megaco
- *Group of signalling poin-to-point:* include H.323 and SIP

H.323 is deployed by ITU_T, intergrated with many difference protocols corresponding to each phase of signalling process. So, this protocol becomes bulky, complexly standardized and suitable only to the terminations with big capacity to store the set of instruction. Futhermore, protocols intergration module makes itself difficult to expand and there is not much potentialities to develop in the future.

In this curcumstance, SIP protocol, developed by IETF, deployment is started later and hasn't standardized clearly yet, however, is seemly paid attention. Now, SIP is accepted by 3GPP and many popular providers. Main element for this advantage is from essence of characteristic: at the beginning, SIP is formed to be able to expand but remain also the simplicity. This protocol has the text UTF-8, inherited two Internet protocol HTTP and SMTP, may take advantage and promote wonderful capablities brought from Internet. An application layer protocol, SIP not only invites more participants to joint the existing sessions (multicast conferences) but also enable to add or remove media from the available session. SIP supports name mapping and forwarding services, since then supports the personal mobility, which means that user can guarantee the only determination visualy without knowing their network location.

Futhermore, with module architecture, SIP is not a vertical-intergrated communication system simply. In fact, it is a component simply used by other protocols of IETF to build a complete multimedia system. Especially, this architecture includes such protocols as real time protocol (RTP), media gateway control protocol (MEGACO) and session description protocol (SDP), RFC 2327. So, SIP should be used with the others protocols to make the perfect services for user, however its functionality and operations do not depend on these protocols.

Thus, this protocol can be applied to many different terminations, from the big-capacity one to small portable devices with very modest capacity. Though the comparison between two protocols must depend on the specific situation and we can't withdraw the final conclusion, it is predicted that SIP is the principal for the development of next generation network (NGN) later.

1.3. Benefits of SIP

Simplicity: SIP stack is smaller than any other VoIP protocol. SIP can be considered a simple tool allowing intelligent terminations, gateways, processes and clients to be constructed and realized.

Scalability: peer – to – peer architecture allow the extension with not high cost. In the comparison to other protocols, requirements for hardware and software to add user into one system decrease considerably.

Functional distribution: SIP allows multiple functions in each component. Changes with a specific component give a small influence to the rest of the system.

2. BASIC CONCEPTS ABOUT SIP

2.1. SIP messages

The elements participating in a SIP-based system will signal between each other by exchanging SIP messages. This is the most basic and visual concept of this protocol. By analyzing the message structure, we can control as well as manage calls. SIP messages are text-based messages and have the general structure as shown below:

- A start line.
- One or more header fields.
- One empty line, followed by an optional body.

There are two types of SIP message: request and response. These two types of message have totally different functions. While requests are sent in order to tell the receiving element to execute a particular task, responses are return to report the result of executing that task.

Structural, these two types of message have different start line. For requests, the start line shows the method of request and the URI to which the request is being sent. For responses, the start line just shows the status code of the response without URI because the respone will go on the same path as the received request.

The header fields carry necessary information for forwarding the message (such as Via, Route, Record-Route…) as well as for processing the message. There are header fields that just appear in requests, or just appear in respone as well as those that appear in both types. A table of details is present in [8], [9].

The body of messages can be present or not. When it is present, its type is described by the header field Content – Type. This body has different significances depending on its type. For different application of SIP, this body will be processed differently. For instance, the body can carries a photo of the participant or necessary session information [8].

Usually, in session establishment process, relevant participants will exchange session information in the body of SIP messages. This session information is described using some session descriptions, such as the Session Description Protocol (SDP) [10].

2.2. SIP elements

The main elements participating in a SIP – based system consist of two types:

- ❖ Network access elements or end – point devices: these are elements that interact directly with users to establish calls, or servers that provide content. In SIP – terminology, they are called User Agents (UA) [8]. [9].
- ❖ SIP – network core elements: these are intermediate elements participating in forwarding SIP messages. There are three main types: proxy server, registration server (registrar) and redirect server.

User Agents can be a PC softwares (softphones) as well as embedded devices (SIP phones). These UA can act as two logical roles: when receiving requests and returning responses, they act as User Agent Server (UAS); when generation requests to send and processing respone, they act as User Agent Client (UAC). For more details about main behaviors of UA, readers can refer to section 8 of [8].

Proxy servers are elements that directly forward SIP messages. Proxy server do that task based on a database called location service. This database associates SIP addresses with particular host addresses that the users are being logoned. This database is constructed by registrars through registration process.

Registration process is realized by REGISTER requests sent to registrar. When a UA want to register its SIP address with a registrar, it generates a REGISTER request and sends to registrar. Registrar receives and processes this request, save into the database bindings between this SIP address and host addresses. One user can login on many hosts. UAs will periodically update information by re-send REGISTER requests to registrar. If after some expired time, registrar does not receive any REGISTER request, it will remove the corresponding items from location service. For more details on registration process, please refer to [8].

When a SIP – based system is expanded, they make use of redirect servers. The operation of redirect server can be generally described as followed: redirect servers just receive SIP requests, and look up location service to generate a 3xx response and send this response to the origin of the request. The sender of the request then re-sends the request to the new addresses shown in this 3xx response. As a result, instead of directly forwarding requests like proxy servers, redirect servers push this task to senders of requests and leverage the processings at network core [8].

Details on processing in different SIP elements are described in [8].

2.3. Structure of SIP protocol

General structure of SIP protocol consists of three layers.

- ❖ The top-most layer is the Transaction User layer (TU). This is the main processing part in SIP elements like UA core, proxy core.
- ❖ The next layer below TU is Transaction layer. This layer does the task of sending and receiving SIP messages reliably. When SIP is run on a unreliable transport layer such as UDP, this layer will retransmit messages arcording to the finite state machines in section 17 of [8].
- ❖ Below the Transaction layer is the Transport layer. SIP is a protocol that can be run on many different transport protocol, unreliable (like UDP) or reliable (TCP, SCTP), as well as secured transport (TLS over TCP)…

Moreover , there are some other concepts like transaction, dialog.

A transaction consists of one request sent by User Agent Client and all responses received for that request. On notable case is INVITE request[8].

A dialog represents a relationship between to UA during some time. Dialogs are included in one session. For multi-party sessions, there can be more than one dialog, one relates to a pair of two parties. The concept of dialog is to make it easy in routing SIP messages.

From that concept, we have the concept of mid-dialog and out-of-dialog requests. A mid-dialog request is sent by a UA to another UA where these two UAs have established a dialog before. In contrast, a out-of-dialog request is sent by a UA to another UA that is not related to any dialog. For instance, INVITE request to establish call with another UA is a out-of-dialog request.

3. SYSTEM IMPLEMENTATION AND THE APPLICATION PROGRAMS

3.1. Model of system

3.1.1. Hardware

Model as figure 2 implement a simple SIP system including two SIP UAs and one SIP Proxy. These three components run on three different computers, connecting to each other in a LAN through a Hub and Ethernet cards.

Hardware requirements

Two computers taking the role of UA must be equipped with sound card, microphone and speaker to support voice communications.

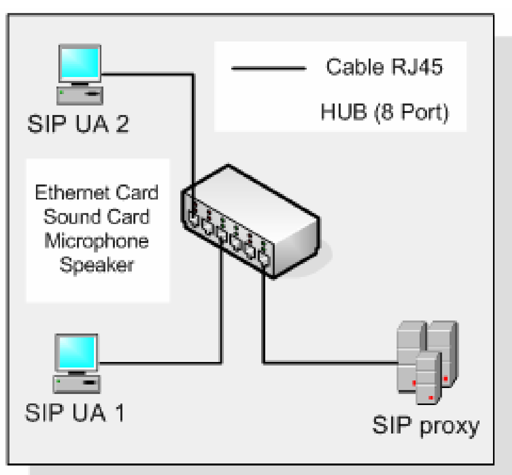

Figure 2: Simple model of system

This model is chosen because it is typical SIP model in a small region, here is a small-size and medium-size enterprise network. In such a network, redirection a message to other region won't exist. From that, Redirect Server's functions almost unconsiderable. We just examine the presence of Proxy Server. The last server, Registrar Server is intergrated to Proxy's functionalities as the logical functions. Moreover, database for such a medium region is not big worthing the utilisation of a private server to store which if used, it cause a waste of resources.

This is also simple model, easily deployed and experimental, tested. After the success of this model, the construction of local enterprise network system will become easy because it is in fact the extension of this model. To add a computer in the system, with the matching to hardware requirements, it can be connected to the network through Hubs.

With such an installation of system, next requirement to enable a voice communication is the following Call Flow of system, in which message is transmited and received continuously.

3.1.2. Call Flow

Figure 3 describe processes transfering messages in a call using SIP, typically over 16 steps.

Step 1 – User at UA1 want to connect to the callee. At that time, software running on UA1 generates an INVITE and send it to proxy. Proxy address is configed in advance in the program.

Step 2 – Proxy receives this request by listenning on port 20050/UDP. It responds immediately with a response 100 (TRYING) requiring client wait the connection. Then, goes next to the step 3.

Step 3 – Proxy checks the INVITE request, determines host at which callee did login, forward INVITE to that host.

Step 4 – UA2 responds to proxy with the response 100 (TRYING).

Step 5 – UA2 sends response 180 (RINGING) to proxy to inform the ringing status.

Step 6 – proxy forward 180 (RINGING) to UA1.

Step 7 – UA1 receives 180 (RINGING), immediately generates the ringback tone formed as the available configuration for color ring.

Step 8 – Callee at UA2 hook-off. UA2 informs to proxy the hook-off signal with 200 (OK).

Step 9 – proxy forwards 200 (OK) to UA1.

Step 10 – UA1 sends request ACK to proxy confirming that it received 200 (OK).

Step 11 – proxy forward ACK to UA2

Step 12 – UA1 generate a RTP stream with UA2 and begins to transmit voice.

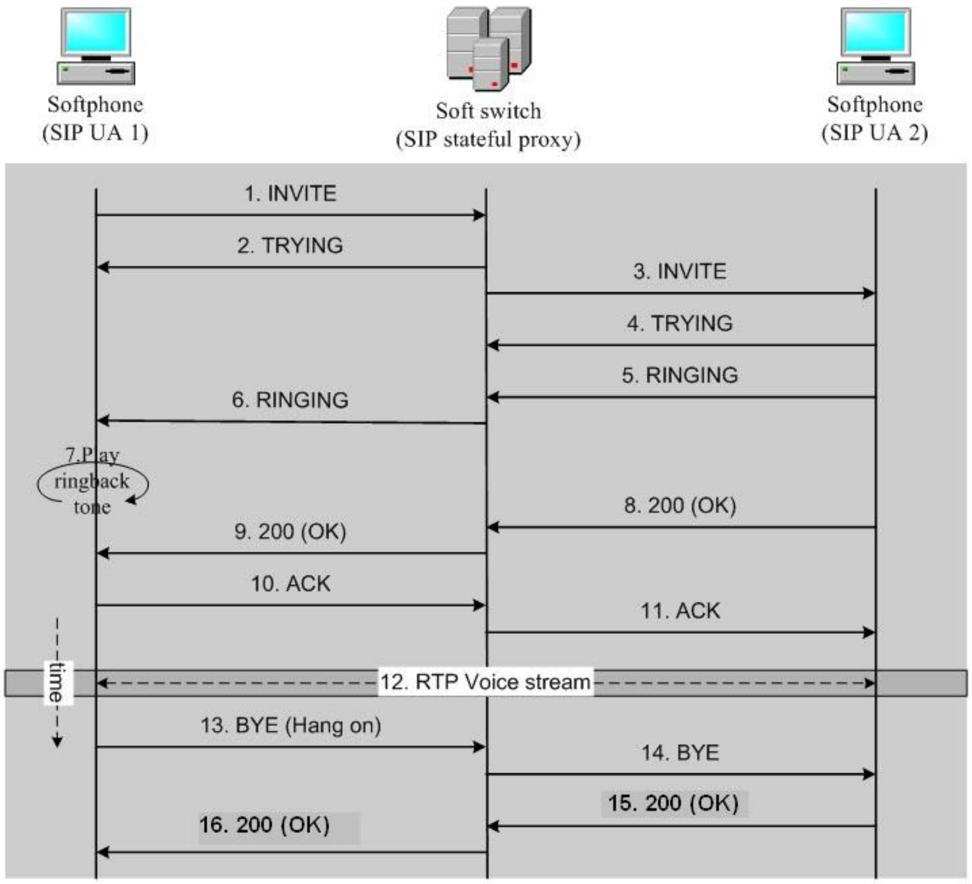

Figure 3: Call Flow of system

Step 13, 14 – Assumped that caller at UA1 hook-on before. UA1 generate request BYE and sens it to UA2 via proxy.

Step 15, 16 – UA2 receive BYE request of UA1, it responds by 200 (OK), terminate the call.

In fact, in the real operations of system, there are many other different situations, such as one in two PC want to terminate the call in the signalling process, it can sends a BYE request or CANCEL to the rest one. Upper situation is just the typical operation of system.

Requests and responses of SIP applied in the system

Table 1. Requests used

INVITE	Inform to begin to initiate the session (of used to modify the session parameters)
ACK	Confirm to receive the final response.
BYE	Terminate the session (cussessful).
CANCEL	Cancel waiting-established session.
REGISTER	Register address which user login to proxy

Figure 2. Responses used

1xx code	100	Continue (Trying)
	180	Ringing
2xx-6xx code	200	OK
	400	Bad request
	403	Forbidden
	408	Request time-out
	600	Busy
	603	Decline
	604	Does not exist

3.1.3. Software

There are two programs realized for this system to run on two type of computers. First one is SIP_UA running on two communication computers (which do the voice transmission to each other), the rest one runs on computer taking the responsibility of proxy, plays an intermediary role for call-control.

3.1.3.1. SIP_UA program

This program runs on the computer which is able to communicate voice to each other, in which, each computer can be UAC or UAS depending on if it sends or it receives a request initiating a call.

Main interface

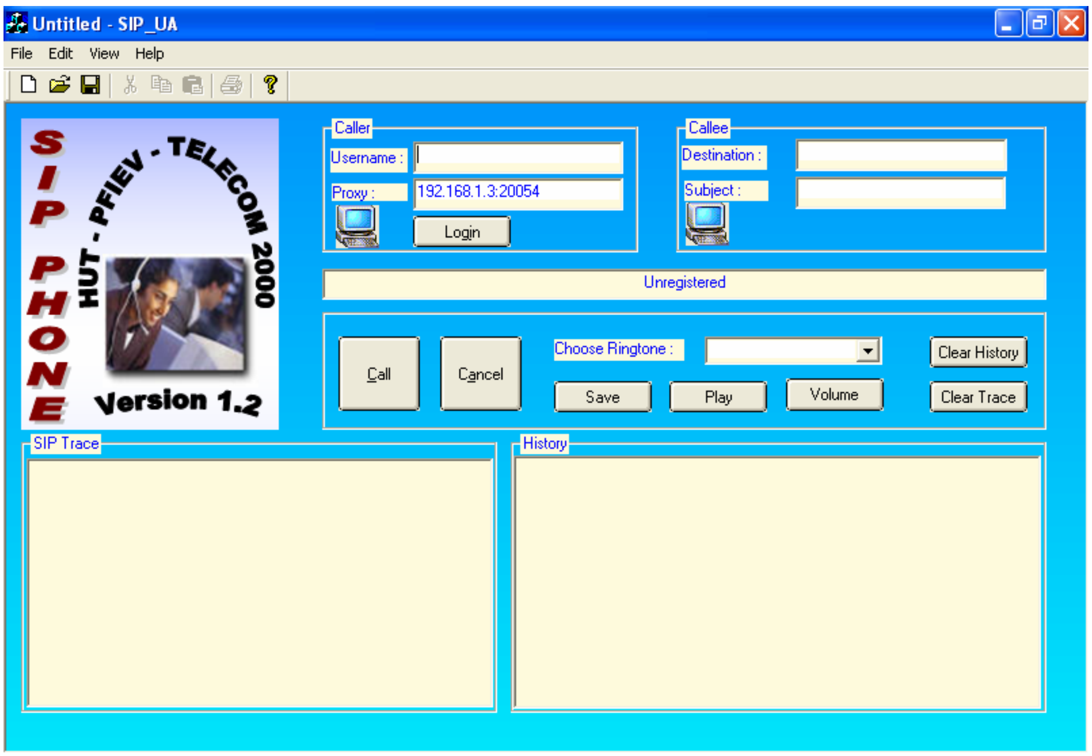

Figure 4: Main interface

✓ SIP_UA (user agent)

- SIP client runs on the Windows 32bit platform.
- Simulation of regular telephone, just be used in local network.
- Speech processing including recording, playback and saving the voice.
- Voice media used is sound card and codec is G.711.
- Signalling port used (using SIP) is 20050/UDP.
- RTP port (using RTP) is 20052/UDP.

First, user must login with proxy using an arbitrary username. Then, he/she needs to determine the IP address and port receiving messages of proxy. These parameters can be changed by correcting a proxy-parameter specification file in the program.

If the login process is successful, status textbox will display the *Idle* text which means that computer has registered in the Idle status and ready to accept any incomming call. On the contrary, user must re-input IP address and port until successful.

To call an arbitrary person, user input the username of callee. This name has the SIP formation: sip:name@host, host is the domain name used by enterprise. If want, user can also input the content of call (optional). Then, push the Call button to initiate the call. In the computer of the callee, there is ringing tone which can be changed as the demand. There is also the ringback tone in the computer of caller. If callee accepts, he/she pushes the Call button, if not, uses the Cancel button to postpone the call.

In the transmission of voice, all SIP messages received will be displayed in listbox SIP Trace. At this point, this program has also the function of simulation SIP protocol, clearly. Just keep trace of process of sending and receiving the messages, we can illustrate an outline of SIP operations.

Another listbox displays list of recent calls. These calls have three main statuses. First one is Dialed, which means that this call is initiated by the caller on this computer, with the name of callee and the date+time of the call. Two remain status are Received, calls received and Missed, calls missed (caller cancel the call before callee accept it), with the same display as above status.

User can adjust the volume of speaker as well as the volume for recording which is the most suitable with user through pushing the Volume button. Final function of the program is the capablity of saving the voice content of all the call to a standard wav form file (Save button), then user can listen again by Play button.

3.1.3.2. SIP-proxy program

✓ **SIP proxy**

- Intermediate element controlling calls between users.
- Listening on port 20050/UDP to receive messages for UAs.
- Implemented in modular structure with many plugins. Extendable by attaching necessary plugins.

When receiving one message on listening port, a necessary plugin is called to analyze the message. If it was a SIP message, this message will be analyzed into a C-liked structure represent SIP message with start line, header fields.. and a new context is created for that request.

The C-liked structure just returned will be processed by hooks attached by plugins. For example, hook attached by syntax plugin will test the syntax of the message… Hooks can be assigned different priorities. One plugin can has many hooks.

After processed by plugins, a set of necessary destinations is created, to which the message will by forwarded. Proxy can be configured in forking or sequential mode. In forking mode, request is forwarded simultaneously to all destinations. In sequential mode, request is forwarded to each destination until succeeded at a particular destination. For each time of forwarding, a new structure is created representing that branch. Finally, proxy receives responses, maybe from many branches, collects response, analyzes and chooses the most appropiate to return. For detail processing, the program follows [8].

4. CONCLUSION AND DEVELOPMENT

This article presents a rather detail view about the SIP protocol, and designs a application voice system based on this. This system can be applied in a small and medium size enterprise. However, the implemented program still lacks some functionalities, including media-types negotiation capability. The program does not support NAT, so it can only operate on LANs.

4.1. Negotiation of media types

In the program, voice transmission process only takes place with data formated in PCMU. There are many types of data encryption and here we only emphasized on popular encrytion type such as:

G.728: CELP, 16 Kbps.
G.729: CELP, 8 Kbps.
G.723: one component in H.323 standard suite, 5,3Kbps, 6.3 Kbps.

The implementation of these encryption types provides many advantages for voice transmission over IP. One obvious example is when one of two parties in a call does not suppot the format requested by the other, it cannot communicate with the other party.

Many format also help the administrator in bandwidth – controlling. If the call takes place during off-peak time, administrator will allow users using high-rate formats. In constrast, administrator will request that users use low-rate format with decreased voice quality.

Thus, one SIP compliant end – point device that supports many format standards is more flexible.

4.2. Video supplementation

One notable issue is the need for visual contact when talking. For that reason, to complete the program, there is a need to supplement video capablity. In addition to visual contact, users can also directly send movies or musics to each other. To make this possible, it is essential to provide more knowledge to capture images from a video recording device or from a multimedia program. Windows supports APIs to do this, similar to APIs for sound. Then, data needs to be encrypted and packaged into RTP packets. The last thing is to created new threads for recording, presenting and saving data. This relates to multithreaded programming techniques used in the program.

4.3. SIP-PSTN interaction

Nowadays, SIP protocol has a vast applications with many types of data, but it is still limitted only on Internet. The conversion of communications systems and Internet is always considered. Most communication on world can be transferred over these systems now. In order to compatible with that trend, there is a need to develop the program to incorporate telecommunications systems. To do this, there are some approaches, for instance, interpretation of signalling message and data across gateways.

Interpretation is a directly mapping between PSTN protocol and SIP. In this approach, popular messages in each protocol are mapped to each other, and a SIP call from a PSTN gateway will not be different from a SIP call from a device.

To to this, first, we have to study signalling protocols in traditional communication systems like ISUP, Q.SIG, message structure in order to packetize data. When data comes across gateways between two different network, it will be interpreted into appropriate format of output network. This function need to be implemented by the program.

With the results received from this development, the program can be applied for both mobile network and different application models such as: PC – to – PC, PC – to – phone or phone – to – phone…

ABBREVIATIONS

ACK	Acknowledgement
API	Application Program Interface
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunication Union
LAN	Local Area Network
MGCP	Media Gateway Control Protocol
PCM	Pulse Code Modulation
PSTN	Public - Switched Telephone Network
RAS	Registration, Admission and Status
RFC	Request for Comment
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transport Control Protocol
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Uniform Resource Indicator
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
WAN	Wide Area Network

REFERENCES

1. Nguyễn Hồng Sơn, *Kỹ thuật điện thoại qua IP & Internet*, Lao Động Xã HộI publisher, TP.HCM, 2003.
2. Alan B. Johnston, *SIP: Understanding the Session Initiation Protocol, Second Edition*, Artech House, 2004.
3. James F. Kurose & Keith W. Ross, *Computer Networking, Second Edition*, Addison Wesley, 2003.
4. Arthus Dumas, *Programming Winsock*, SAMS, 1995.
5. Martin W. Murhammer, et al., *TCP/IP Tutorial and Technical Overview*, IBM, October 1998.
6. Douglas E. Comer and David L. Stevents, Internetworking with TCP/IP, Vol III: Client-Server Programming and Application, Prentice Hall, 1993.

7. Joe Casad, *Sams Teach Yourself TCP/IP in 24 Hours, Third Edition*, Sams, September 03, 2003.
8. Rosenberg, J., et al., *SIP: Session Initiation Protocol, RFC 3261*, June 2002.
9. Handley, M., et al., *SIP: Session Initiation Protocol, RFC 2543*, March 1999.
10. Handley, M., and V. Jacobson, *SDP: Session Description Protocol, RFC 2327*, 1998.
11. Schulzrinne, H., et al., *RTP: A Transport Protocol for Real-Time Applications, RFC 3550*, 2003.
12. Donovan, S., *The SIP INFO Method, RFC 2976*, 2000.
13. Rosenberg, J., *The Session Initiation Protocol (SIP) UPDATE Method, RFC 3311*, 2002.
14. Rosenberg, J., and H. Schulzrinne, *Session Initiation Protocol (SIP): Locating SIP Servers, RFC 3263*, 2002.
15. Schulzrinne, H., *RTP Profile for Audio and Video Conferences with Minimal Control, RFC 3550*, 2003.
16. Vermuri, A., and J. Peterson, *Session Initiation Protocol for Telephones (SIP-T): Context and Architectures, RFC 3372*, 2002.
17. Johnston, A., et al., *Basic SIP Call Flow Examples, IETF Internet-Draft*, Work in Progress, April 2003.
18. *Pulse Code Modulation (PCM) of Voice Frequencies,Recommemdation G.711, Fascicle III.4*.
19. ITU-T Study Group 15, *Annex A: Reduced complexity 8 kbit/s CS-ACELP speech codec, Recommendation G.729 - Annex A*, November 1996.
20. ITU-T Study Group 15, *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, March 1996.