

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES

VOIP OVER WIRELESS NETWORKS

by
Gamze TEKİN

February, 2013
İZMİR

VOIP OVER WIRELESS NETWORKS

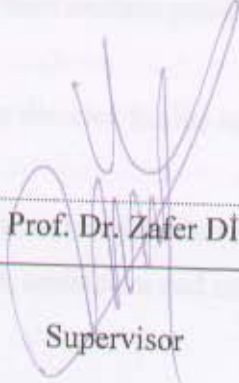
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of Science in
Electrical and Electronics Engineering, Applied Electrical and Electronics
Program**

**by
Gamze TEKİN**


**February, 2013
İZMİR**

M.Sc THESIS EXAMINATION RESULT FORM


We have read the thesis entitled **“VOIP OVER WIRELESS NETWORKS”** completed by **GAMZE TEKİN** under supervision of **ASST. PROF. DR. ZAFER DİCLE** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.


Asst. Prof. Dr. Zafer DİCLE


Supervisor


Prof. Dr. Yavuz SEVİN

(Jury Member)


Prof. Dr. Yalçın GEBİ

(Jury Member)


Prof. Dr. Mustafa SABUNCU
Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank to my advisor Asst. Prof. Dr. Zafer DİCLE for his guidance, assistance and also technical support by providing necessary equipment in order to implement project prototype.

I wish to thank my family for their endless patience and belief in me.

I want to thank to my company director for his understanding and support.

I am also thankful to my husband and collaborate, Mehmet Köse who always facilitates my life with his support, assistance and motivation.

Gamze TEKİN

VOIP OVER WIRELESS NETWORKS

ABSTRACT

Voice over IP and wireless are revolutionary technologies by all means of modern time which change the attributes of communications dramatically. VoIP is simply the transmission of voice traffic over IP-based networks. VoIP has become popular largely because of the cost advantages to consumers over traditional telephone networks whereas Wireless communications is a rapidly growing segment of the communication industry, with the potential to provide high-speed high-quality information exchange between portable devices located anywhere in the world. Since both technologies have shown their existence in today's communication industry individually, merger of these technologies was necessary and hence both technologies are being deployed.

In this thesis, the VoIP technology is examined by regarding its general structure, fundamental components and operation logic. In addition to the detailed explanation of VoIP procedure, the simple VoIP prototype in wireless networks was implemented. Codecs, signaling protocols, real time protocols and media gateway protocols are the main principles of the VoIP technology. These principles and the components such as end systems, signaling servers and media gateways are used to define and implement VoIP process.

A VoIP Phone system requires the use of VoIP phones. VoIP phones come in several types. In this prototype, software based phones (Soft Phones) are preferred. Sipdroid application which runs in Android mobile phone and Peers application which runs in PC are used as SIP clients. These applications use G711 codec technology, SIP signaling and RTP real time protocols. The SIP accounts which are necessary to run these applications are obtained from a free signaling server called as Sip2Sip. This signaling server is responsible from setup of the SIP session establishment and control of the routing of signaling messages.

In order to provide wireless network access to each SIP client, Cisco access points are used in this prototype.

Keywords: VoIP, codec, signaling protocol, real time protocol, Sipdroid, Peers, Sip2Sip, access point.

KABLOSUZ AĞLARDA İNTERNET PROTOKOLÜ ÜZERİNDEN SES İLETİMİ

ÖZ

IP üzerinden ses iletimi (VoIP) ve kablosuz ağ, iletişimin özelliklerini ve gelişim yönünü önemli ölçüde değiştiren, günümüz modern zamanın her yönüyle devrimci teknolojileridir. VoIP, IP tabanlı ağlar üzerinden ses trafiğinin basitçe iletimidir. VoIP teknolojisi, geleneksel telefon ağları kullanımına kıyasla ücretlendirmede oldukça avantaj sağladığı için kısa zamanda fazlasıyla yaygınlaşmıştır. Bununla birlikte, kablosuz ağlarda haberleşme, dünyanın herhangi bir yerinde bulunan taşınabilir cihazlar arasında yüksek hızda ve yüksek kalitede bilgi değişimi sağlama potansiyeline sahip olduğu için haberleşme endüstrisinin hızla büyüyen bir sekmendi olmuştur.

Bu tez çalışmasında, internet protokolü üzerinden ses iletimi (VoIP) teknolojisi, genel yapısı, temel bileşenleri ve çalışma mantığı göz önüne alınarak incelenmiştir. VoIP prosedürünün detaylı anlatımına ek olarak, kablosuz ağlarda basit bir VoIP prototipi gerçekleştirilmiştir. Kodlama/kod çözme teknikleri, sinyalleşme ve gerçek zamanlı iletim protokolleri, VoIP teknolojisinin temel prensipleridir. Bu prensipler ile sinyalleşme sunucuları, medya ağı geçitleri ve uç noktalarda bulunan elektronik cihazlar gibi sistem bileşenleri, VoIP sürecini tanımlamak ve gerçekleştirmek için kullanılmaktadırlar.

Bir VoIP telefon sistemini gerçekleştirmek için bu sisteme uygun olarak çalışabilecek VoIP telefonlarına ihtiyaç vardır. VoIP telefon birkaç farklı çeşitte olabilir. Bu prototipte, VoIP telefon çeşitlerinden yazılım tabanlı VoIP telefonlar tercih edilmiştir. Android akıllı telefonda çalışan Sipedroid uygulaması ve PC’de çalışan Peers Java uygulaması, SIP alıcıları olarak kullanılmıştır. Bu uygulamalar, G711 kodlama/kod çözme teknolojisini, SIP sinyalleşme ve RTP gerçek zamanlı iletim protokolünü kullanmaktadır. Uygulamalar için gerekli olan SIP hesapları, Sip2Sip olarak adlandırılan ücretsiz bir sinyalleşme sunucusundan elde edilmiştir. Bu

sinyalleşme sunucusu, SIP oturumunun kurulmasından ve sinyalleşme mesajlarının yönlendirilmesinin kontrol edilmesinden sorumludur.

Her bir SIP alıcısının kablosuz ağı erişimini sağlamak amacıyla Cisco erişim noktaları bu prototipte kullanılmıştır.

Anahtar sözcükler: VoIP, PSTN, kodlama/kod çözücü, sinyalleşme protokolü, gerçek zamanlı iletim protokolü, Sipedroid, Peers, Sip2Sip, erişim noktası.

CONTENTS

	Page
THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZ	vi
 CHAPTER ONE – INTRODUCTION	1
1.1 Introduction	1
1.2 Historical Perspective.....	2
1.2 Literature Overview	4
1.3 Thesis Outline.....	7
 CHAPTER TWO – (VOIP) VOICE OVER IP TECHNOLOGY.....	8
2.1 VoIP Structure	8
2.2 How VoIP Works?	9
2.3 VoIP Technology Components	13
 CHAPTER THREE – VOIP PRINCIPLES.....	17
3.1 Codecs	17
3.1.1 G.711	17
3.1.2 G.723	18
3.1.3 G.729	18
3.2 VoIP Protocols	19
3.2.1 Signaling Protocols.....	20
3.2.1.1 H.323.....	21
3.2.1.1.1 H.323 Components.....	21

3.2.1.1.2 H.323 Protocols.....	23
3.2.1.1.3 H.323 Call Scenarios.....	25
3.2.1.2 SIP (Session Initiation Protocol).....	26
3.2.1.2.1 SIP Components.....	27
3.2.1.2.2 SIP Protocols.....	29
3.2.1.2.3 SIP Call Scenarios.....	30
3.2.1.3 Comparison between SIP and H.323	30
3.2.1.4 MGCP (Media Gateway Control Protocol)	34
3.2.1.5 Megaco/H.248.....	35
3.2.1.6 Comparison between MGCP and Megaco/H.248.....	35
3.2.2 Real Time Protocols	36
3.2.2.1 RTP (Real Time Transport Protocol).....	37
3.2.2.2 RTCP (Real Time Control Protocol)	39
3.2.2.3 RTSP (Real Time Streaming Protocol).....	41
3.2.2.4 RSVP (Resource Reservation Protocol)	43
CHAPTER FOUR – SIP OPERATIONS	45
4.1 Introduction	45
4.2 SIP Messages.....	45
4.3 SIP Session Establishment	50
4.4 SIP Presence Scenario	52
CHAPTER FIVE – VOIP PROTOTYPE.....	55
5.1 Introduction	55
5.2 Prototype Design	56
5.3 Prototype Components	58
5.3.1 Sipdroid	58
5.3.2 Peers.....	62
5.3.2.1 Architecture.....	62
5.3.2.2 SIP Package Details	64

5.3.2.3 SDP Package Details.....	69
5.3.2.4 Media Package Details.....	70
5.3.2.5 RTP Package Details.....	71
5.3.2.6 GUI Package Details.....	71
5.3.3 Wireless Access Points	73
5.3.3.1 Cisco Aironet 1130AG Series Access Point.....	74
CHAPTER SIX – CONCLUSION	78
6.1 Conclusion.....	78
6.2 Future Works	80
REFERENCES.....	81

CHAPTER ONE

INTRODUCTION

1.1 Introduction

Constructing a VoIP telephony service over a wireless IP network requires understanding of VoIP technology and the unique characteristics of the wireless medium.

Wireless LANs (WLANs) are being more and more widely deployed at present, since the number of mobile users is increasing steadily. WLANs are a key element in any business environment where “anytime, anywhere” access to network resources is vital.

First of all the bandwidth available in WLANs is significantly lower than in the case of fixed LANs. For the most widely-spread wireless networks, the maximum theoretical rate is either 11 Mb/s or 54 Mb/s. These rates are considerably lower than the current extensively-used 100 Mb/s and 1 Gb/s fixed LANs. Another difference between the wired and wireless networks is that in wired networks the last part of the connection (from the LAN switch to the PC, for example) is dedicated to one user. However in WLANs the medium is not only shared between the applications of one user, but between all the applications of all the users that happen to be using the same access point at the same moment of time. Hence network quality is more prone to degrade significantly (Beuran, 2006).

Voice over IP (VoIP), also known as Internet telephony, is a form of voice communication that uses data networks to transmit audio signals. When using VoIP the voice is appropriately encoded at one end of the communication channel, and sent as packets through the data network. After the data arrives at the receiving end, it is decoded and transformed back into a voice signal. Many enterprises consider replacing traditional PBX phone systems with a VoIP telephony server. PBX costs may be prohibitive for the new companies that need to set up a telephony system

from scratch. On the other hand, VoIP systems require in principle no significant specific running costs, since they use the same network infrastructure that already exists and is maintained. Using VoIP on wireless LANs solution enables support of mobile devices within the building or campus (Beuran, 2006).

The aim of this study consists of a combination of the two intermediate objectives. The primary objective is to analyze VoIP technology with its structure, components and principles. The secondary objective is to investigate today's widely used VoIP applications with regarding operating system service, signaling protocols and network types. The ultimate objective of this study is to implement a VoIP system prototype over wireless networks. The main issue about this prototype is to make VoIP call over wireless networks between two different VoIP soft phone applications which have different operating system service and run on different platforms.

1.2 Historical Perspective

The global evolution of the Internet and the wide spread growth of networks have been made the Internet part of our everyday life. This is the reason why the interest and demand on different applications has been increased. The raise in demand has produced many new applications. Voice over Internet Protocol (VoIP) technology has become a potential alternative to and supplement of the traditional telephony systems over the Public Switched Telephone Network (PSTN), providing a versatile, flexible and cost-effective solution to speech communications. Basic differences between VoIP calls and PSTN calls are shown in Table 1.1.

Internet telephony is a revolutionary technology that has the potential to completely rework the world's phone systems. Internet telephony is the transmission of voice signals from one party to other party digitally i.e., usage of packet switched data network (PSDN). The first documented internet telephony experiments were conducted on the ARPANET (the forerunner of the Internet) by researchers at MIT in the mid-1970s, resulting in the publication of an Internet protocol specification, RFC741, for the 'Network Voice Protocol', in 1977 (Latif & Malkajgiri, 2007).

Table 1.1 Comparison of quality of voice over PSTN and over IP (Iqbal & Cheema, 2009)

Concept	Voice over PSTN	Voice over IP
Switching	Circuit switching (end to end dedicated link)	Packet switching
Bit Rate	64kbps per 32kbps	14 kbps with overheads (only when talking)
Latency	Lesser than 100ms	200-700ms depending on total traffic on IP network.
Bandwidth	Dedicated	Dynamical allocated
Cost of access/billing	Business customer. Monthly charge for line, plus per minute charge.	Business customer. Cost of IP infrastructure, Hybrid IP/PBX and IP Phones.
Equipment	Dump terminal (Less expensive) intelligence in network	Integrated smart programmable terminals(expensive) intelligence not in network
Quality of service	High(extremely low loss)	Low and variable, but traffic is sensitive depending on packet loss and delay experienced.
Network availability	99.999% up time	Level of reliability not known.
Security	High level of security because of dedicated link.	Possible eavesdropping at router.

These experiments resulted in audio transmission on packet networks but they were limited to academic environments only. As computers of that age did not have the power to compress the audio data below 64kbps or 56 kbps and sound input and

output devices have also to be made because there were none to be bought. But later when the computing power the compress the speech below 14.4 kbps by 1993, the first commercial Internet phone Application appeared (Latif & Malkajgiri, 2007).

The public switched telephone network (PSTN) has been evolving ever seen since Alexander Graham Bell made the first voice transmission over wire in 1876. In traditional telephones, devices are limited to communicating with those devices, which are connected directly, and the telephony companies and their protocols must handle all location and routing features. Traditional telephone uses circuit networks (Latif & Malkajgiri, 2007).

1.3 Literature Overview

An emerging trend for implementing VoIP is in wireless networks. A wireless LAN (WLAN) is a data transmission system designed to provide location-independent network access between computing devices by using radio waves rather than a cable infrastructure. WLANs give users wireless access to the full resources and services of the LAN across a building or campus environment. There are some fundamental concerns that WLANs introduce. These issues include a higher frequency of dropped packets, larger latency and more jitter (Udani & Mehta, 2001).

There are numerous benefits of utilizing WLANs. In any network environment, users would be able to access the network far beyond their personal desktops, giving these mobile users much-needed freedom in their network access. Specifically, they can access information from anywhere in the building or campus. A WLAN system provides a powerful combination of wire line network throughput, mobile access and configuration flexibility. It liberates users from tethered access to the network backbone, given them anytime, anywhere network access. Applications include VoIP from mobile personal communications devices (Udani & Mehta, 2001).

Today, there are many VoIP applications that provide VoIP service over wireless networks. A VoIP phone system requires the use of special phones which are suitable

for VoIP applications. VoIP phones come in several versions/types such as soft phones, hard phones and USB phones. In the scope of this thesis, VoIP softphones are examined.

A softphone is a software program for making telephone calls over the Internet using a general purpose computer, rather than using dedicated hardware. Often a softphone is designed to behave like a traditional telephone, sometimes appearing as an image of a phone, with a display panel and buttons with which the user can interact. A softphone is usually used with a headset connected to the sound card of the PC, or with a USB phone. To communicate, both end-points must have the same communication protocol and at least one common audio codec. Most service providers use a communication protocol called SIP (Session Initiation Protocol) by IETF, except Skype which is a totally proprietary system and Google Talk which is based on Jabber, now known as XMPP (İsmail, 2011).

There are numerous studies that analyze VoIP service over wireless networks by taking care into different aspects such as performance, quality and cost. But, there are few studies that implement VoIP system and analyze VoIP softphone applications.

In one of these studies by Mohd Nazri İsmail, “Analysis of VoIP Softphone Performance between Wired and Wireless in Campus network Environment”, (İsmail, 2011) VoIP system prototype has implemented over wired and wireless technology using softphone in campus network environment. They selected two softphones and plans to use VoIP communications, 3CX softphone and Mizuphone softphone. They measured and analyzed the 3CX softphone and Mizuphone performance during VoIP communication over wired and wireless technology. 3CX softphone achieved a good performance results and selected in order to use for VoIP communication in campus environment. After this study, VoIP was also gaining popularity in the consumer space thanks to the availability of free PC-to-PC calling with softphones such as Skype.

In the another study by G. H. Khaksari, A. L. Wijesinha, R K. Karne, Q. Yao and K. Parikh “A VoIP Softphone on a Bare PC”, (Khaksari, Wijesinha, Karne, Yao, Parikh), the architecture, design and implementation of a VoIP softphone that runs on a bare Intel-386 based PC are described. The performance of bare PC and WinRTP softphones on the Internet are compared by determining call quality and measuring the values of jitter, delay and packet loss. According to this study, a bare PC-to-bare PC connection is associated with smaller values of jitter than a WinRTP to bare PC connection even for larger voice packet sizes. A bare PC softphone also provides better call quality than a WinRTP softphone under heavy system load conditions on a LAN.

In this study, first of all, general information about VoIP technology, which includes VoIP working principles, VoIP components and protocols are given. The superiors and deficiencies of VoIP protocols are states by comparing them with each other. Call scenarios which belong to each protocol are explained in detail in order to understand call process in real VoIP applications. After the general information, today's common VoIP softphone applications are searched and examined. These applications are categorized according to supported operating system services, signaling protocol. Two different open-source SIP Softphone applications are decided to use in the VoIP prototype. In other studies, VoIP call is made over wireless networks between the same VoIP applications by using same operating system service. In this study, two different VoIP soft phone applications are used and VoIP call is made between these two applications.

Finally, the VoIP prototype over wireless networks is implemented. In this prototype, two Cisco access points are used in order to provide wireless network node for each SIP client. In WLANs where more access points are simultaneously active, roaming issue is taken into account. When a node moves or reception conditions change, it will usually select the access point in its range that has the highest signal strength.

1.4 Thesis Outline

This thesis is organized in six chapters. The first chapter covers the literature review about basics of the VoIP technology, historical perspective of VoIP and the aim of this thesis. The remainder of this thesis is organized as follows.

The second chapter deals with the structure of VoIP technology. The components in this technology are specified by emphasizing their functions in the VoIP procedure. Also, the overall working of VoIP technology is explained step by step.

In the third chapter, the principles of VoIP technology are explained in detail. These principles include the most common Codec techniques and VoIP protocols. The superior and deficiencies of the protocols are specified by comparing with each other.

In the fourth chapter, the call operations and the messages in SIP protocol, which is currently the most widely used common signaling protocol in VoIP applications, are examined. The mentioned scenarios include SIP registration and SIP session establishment operations. These operations are given as examples and examined in detail since they got involved also in the prototype.

The fifth chapter includes today's widely used VoIP applications and the VoIP prototype which are designed and implemented in the scope of this thesis. The components which are used in the prototype and the design features are specified.

Finally, the last chapter is the conclusion part of the thesis. The results of the thesis are discussed and the future works are specified.

CHAPTER TWO

VOIP (VOICE OVER IP TECHNOLOGY)

2.1 VoIP Structure

VoIP is one of the most common and cheap technology to communicate short and long distance. It transmits the digitized voice data over IP network which provides a user to have a telephonic conversation over the existing Internet; this voice signal is appropriately encoded at one end of the communication channel transmitted using IP packets, and then decoded at the receiving end which transformed back into a voice signal.

The simple diagram which is shown in Figure 2.1 can easily illustrate the idea of using VoIP calls. VoIP calls start from a Location A, traverse Router A if it's an IP based call otherwise routed towards PBX box which further placed it to PSTN Voice network. This network switches it back to the destination PBX and then placed it to Location C. Whereas the IP call goes from Router A to Router C by the help of IP WAN DATA; they are diverted to the router and terminate over the destination location.

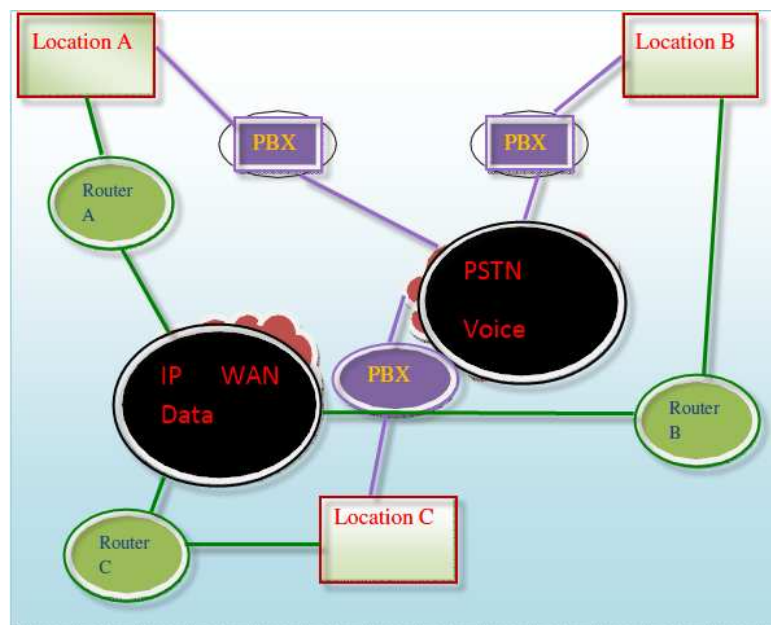


Figure 2.1 Illustration of a VoIP system (Mehdi, 2009).

Figure 2.2 shows the internal structure of VoIP calls made by IP phone in little bit more details, it starts from the IP phone, first user press the digital number on dialing pad which translate these digital numbers into binary codes, these binary codes convert into IP packets and transmits towards the Local Area Network (LAN). They further transmit it towards the router which analyzes the IP address of the destination and transmit further through the IP Network. The call has been treated according to the destination, for instance if it's meant for an ordinary telephony then will be directed towards a PSTN Gateway which further switches towards the right destination. But if it's a VoIP call then it will go to the relevant router which analyzes the IP address and direct towards a relevant LAN and then which further could be attended by an IP phone or a soft phone (software in computer or in a phone).

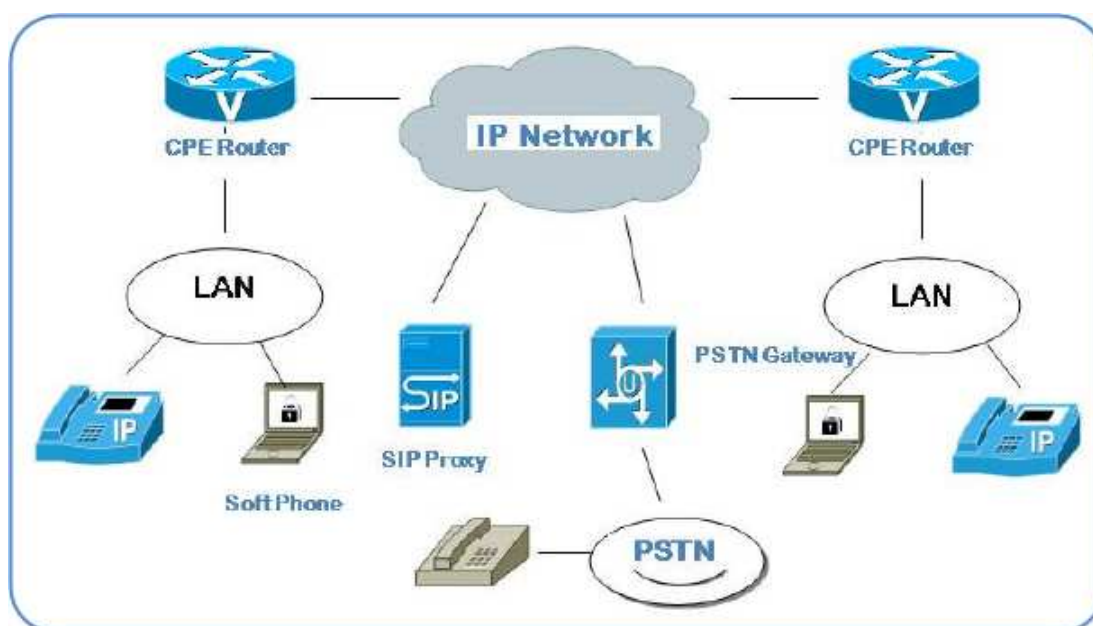


Figure 2.2 VoIP internal structure (Mehdi, 2009).

2.2 How VoIP Works?

VoIP uses Internet Protocol for transmission of voice as packets over IP networks. The process involves digitization of voice, the isolation of unwanted noise signals and then the compression of the voice signal using compression algorithms/codecs. After the compression, the voice is packetized to send over an IP network. Each

packet needs a destination address and sequence number and data for error checking. The signaling protocols are added at this stage to achieve these requirements along with the other call management requirements. When a voice packet arrives at the destination, the sequence number enables the packets to be placed in order and then the decompression algorithms are applied to recover the data from the packets. Here the synchronization and delay management needs to be taken care of to make sure that there is proper spacing. Jitter buffer is used to store the packets arriving out of order through different routes, to wait for the packets arriving late (Bakshi, 2006). There are many intermediate devices which serve the purpose as shown in the Figure 2.3.

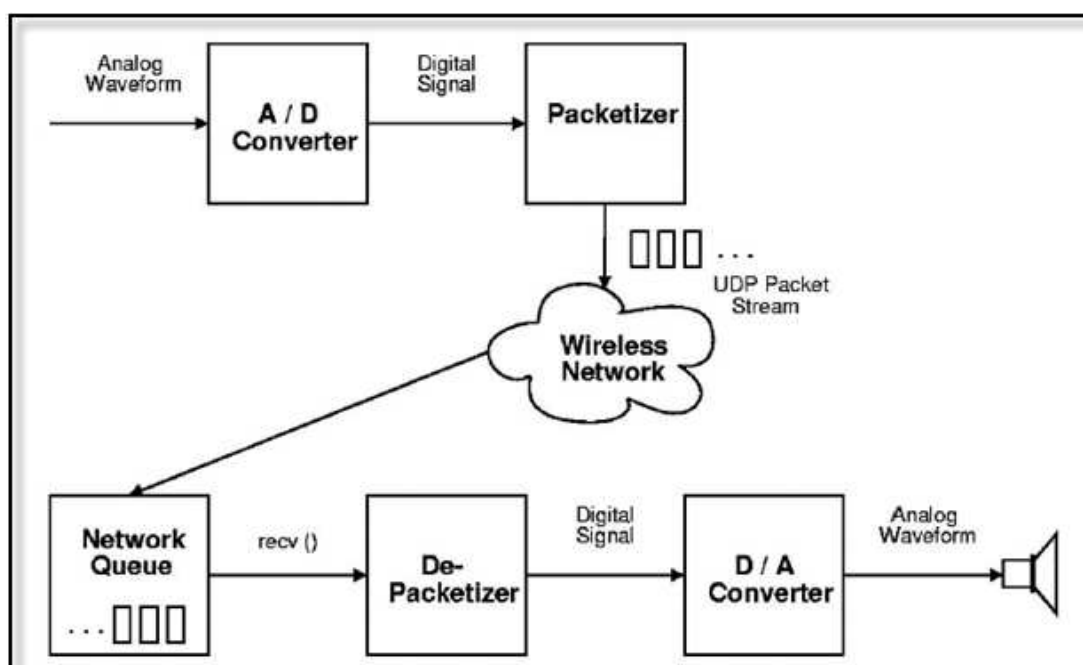


Figure 2.3 VoIP process (Iqbal & Cheema, 2009)

The overall working of VoIP is summarized at the following steps.

- **Voice Capture:** VoIP uses Internet Protocol for transmission of voice as packets over IP networks. VoIP communication needs an audio input device, like in ordinary PSTN system, such as a microphone, to send the audio signal. An analog-to-digital converter is used to transform that audio signal into digital bytes packets.

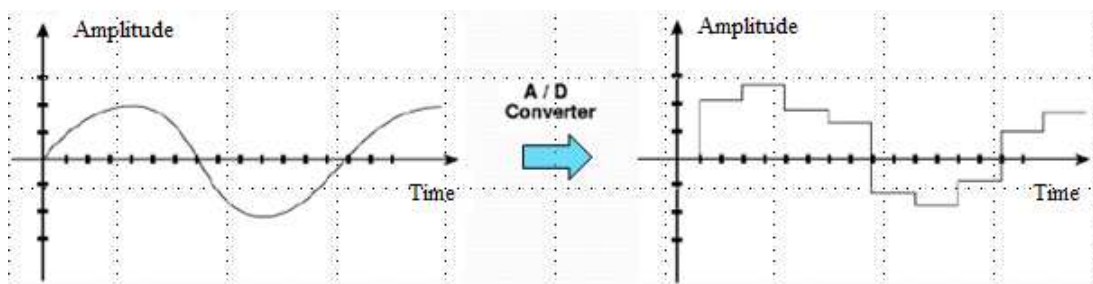


Figure 2.4 Analog to Digital conversion

- **Audio Data Encoding:** Before sending the digital signal it is important in packet-switched networks to prioritize voice data to be encoded. Then speech compression is engaged at this stage. Traditional telephone networks use pulse code modulation (PCM) at 8K samples per second. 12-bit samples are compressed and expanded by a nonlinear look-up table into 8-bit words giving a transmitted rate of 8kbit/s. The compression typically used by an Internet phone today is of the order of 16 to 1 (128kbit/s to 8kbit/s). Such compression is beyond PCM, ADPCM (32kbit/s, used in CT-2 cordless phones), or sub-band coding (down to 16kbit/s for speech bandwidths, normally used for music at higher bit rates). In case of a LAN (local area network) when there is sufficient bandwidth there is no need of compression (Latif & Malkajiri, 2007).
- **Packetization:** After the compression, the voice is packetized to send over an IP network. The first packetization is implemented at application level by using RTP protocol. The voice packets are converted into data packets with RTP protocol. RTP data packets are send to transport layer.
- **Transport Layer (UDP):** The transport layer provides the rules required for sending the data. Most data travelling over the Internet uses the Transmission Control Protocol (TCP) for the transport layer because it guarantees data delivery and integrity. VoIP does not need the kind of delivery guarantee which TCP provides, so IP network in VoIP transmissions can use an alternative faster transport layer protocol, user datagram protocol (UDP). In

transport layer with UDP protocol, data is transmitted in the form of datagrams. Every datagram has a source address, destination address and sequence number. Each datagram of file/message is independently routed across the network and packets are reassembled at the receiving end.

- **Network Layer (IP):** The data packets are send into Network layer in the form of datagrams. The network layer consists of the IP which establishes a connection between two computers. The Internet Protocol (IP) is provided for routing datagram between any two nodes with checking for corruption and loss.
- **Application Layer:** Once VoIP data arrives at its destination, the application layer interprets it and presents it to the user. In the application layer, Voice over IP (VoIP) uses signaling protocols (H.323, SIP, and MGCP) for establishing connections between endpoints and also, it uses media protocols (RTP, RTCP and RTSP) for dealing with the real time data such as audio or video. The most commonly used application layers for VoIP are SIP and RTP.
- **Signaling:** In the application layer, signaling system has to perform its work and it does the following tasks (Latif & Malkajgiri, 2007).
 1. Try to find out the destination IP address.
 2. After finding destination IP address and it establishes communication with that party.
 3. After negotiating the Internet protocol performs voice compression, buffer length and time stamping of packets and starts communication. However situation becomes more complex if signaling system has to communicate with gateway between the Internet and PSTN. Gateways are devices that allow calls to be placed to and from other telephone networks, which are implemented between Internet and PSTN. Although gateway cannot support the same number of users as even the smallest local telephone exchange. In the case of outgoing calls

VoIP phone captures the phone number and the IP address of gateway. But in the case of reverse direction that is from PSTN to internet it is rather impractical for the PSTN user to enter the telephone number of the gateway and then the numeric IP address of the desired party.

- **Audio Playback:** Finally at the receiving end, packets have to be disassembled for data extraction and for converting the data into analog voice signal and send those signals to the sound card of the respective device.

2.3 VoIP Technology Components

An Internet telephony system contains three types of components: end systems, signaling gateways and signaling servers.

- End systems are electronic devices with which clients or users place and receive calls.
- Gateways are devices that allow calls to be placed to and from other telephone networks.
- Signaling servers handle the application level control of the routing of signaling messages.

An end system can originate a call, it also accept, reject or forward incoming calls. When this end system places a call, the call establishment request can proceed by a variety of routes through components of the network. At first, the originating end system must decide where to send its requests. There are two possibilities here: the originator may be configured so that all its requests go to a single local server; or it may resolve the destination address to locate a remote signaling server or end system to which it can send the request directly. Once the request arrives at a signaling server, that server uses its user location database, its local policy, DNS resolution, or

other methods to determine the next signaling server or end system to which the request should be sent. A request may pass through any number of signaling servers: from zero (in the case when end systems communicate directly) to the entire server on the network (Tong, 2005).

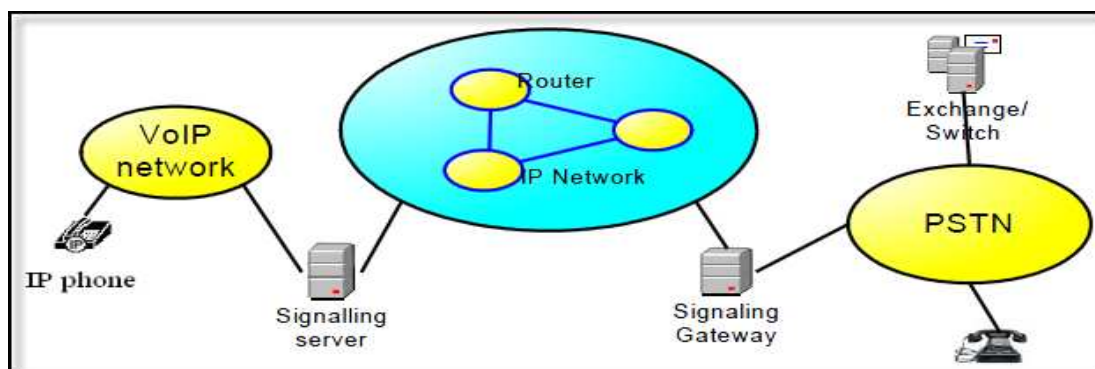


Figure 2.5 Generalized model

A Media Gateway acts as a translation unit between disparate telecommunications networks such as PSTN; Next Generation Networks; 2G, 2.5G and 3G radio access networks or PBX. Media Gateways enable multimedia communications across Next Generation Networks over multiple transport protocols such as ATM and IP. Media gateways, also commonly referred to as VoIP gateways are devices which bridge conventional telephone networks and equipment to VoIP telephone networks. VoIP Media Gateways perform the conversion between TDM voices to Voice over Internet Protocol (VoIP). A typical media gateway has at least one conventional telephone port and at least one Ethernet port (Freeman, 2005).

As the Media Gateway connects different types of networks, one of its main functions is to convert between the different transmission and coding techniques. Media streaming functions such as echo cancellation, DTMF, and tone sender are also located in the Media Gateways (Freeman, 2005).

Media gateways are part of the physical transport layer. They are regulated by a call control function housed in a media gateway controller. A media gateway, with its associated gateway controller, is necessary for the network transformation to

packetized voice (Freeman, 2005). Several of the media gateway functions are listed below:

- Carries out A/D conversion of the analog voice channel (called compression in many texts);
- Converts a DS0 or E0 to a binary signal compatible with IP or ATM;
- Supports several types of access networks, including media such as copper (including various DSL regimes), fiber, radio (wireless) and CATV cable. It is also able to support various formats found in PDH and SDH hierarchies;
- Capable of handling several voice and data interface protocols;
- It must provide interface between the media gateway control device and the media gateway. This involves one of four protocols: SIP, H.323, MGCP and Megaco (H.248);
- It can handle switching and media processing based on standard network PCM, ATM and traditional IP;
- Transport of voice. There are four transmission categories that may be involved:
 1. Standard PCM (E0/E1 or DS0/DS1)
 2. ATM over AAL1/AAL2
 3. IP-based RTP/RTCP
 4. Frame relay

The gateway controller or media gateway controller (MGC) carries out the signaling function on VoIP circuits. Some texts call an MGC a ‘softswitch’, even though they are not truly switches but servers that control gateways (Freeman, 2005). This function is illustrated in Figure 2.6.

An MGC can control numerous gateways, but to improve reliability and availability, several MGCs may be employed in separate locations with function duplication on the gateways they control. Thus, if one MGC fails, others can take over its functions. That is, establishing telephone connectivity, maintaining that

connectivity, and taking down the circuit when the users are finished with conversation (Freeman, 2005).

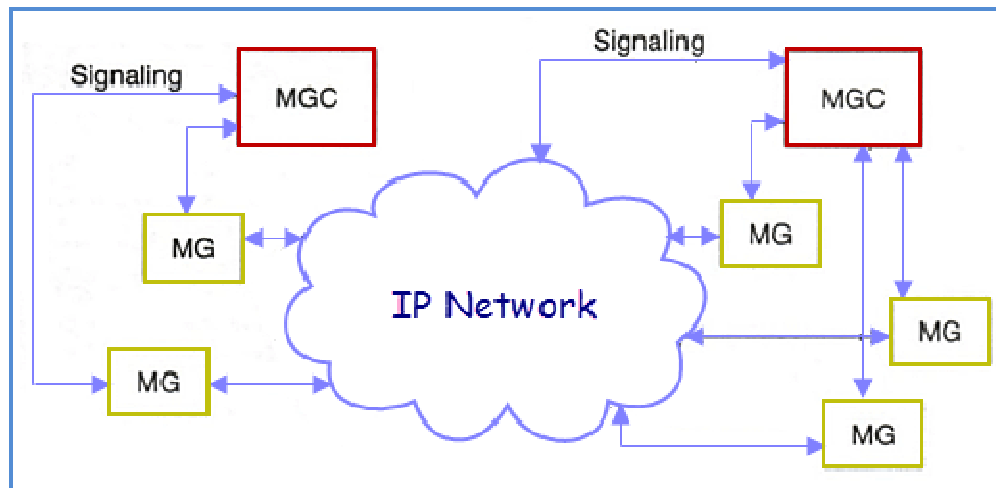


Figure 2.6 The media gateway controller (MGC) provides a signaling interface for media gateways (MGs), thence to the IP network (Freeman, 2005).

CHAPTER THREE

VOIP PRINCIPLES

3.1 Codecs

Compression/Decompression (CODEC) technology has been used in VoIP equipment for converting audio signals into a digital bit stream and vice versa. The main advantage of using compression techniques is that it allows a reduction in the required bandwidth while preserving voice quality in certain degree. There are many compression schemes available but the most VoIP devices uses those CODECs which are standardized by international boards or bodies such as the ITU-T and accepted worldwide for the sake of interoperability across different vendors. Each of them has different properties in relation to the amount of bandwidth it requires, but also, the perceived quality of the encoded speech signal. There are some of the most popular CODECs are G.711, G.723 and G.729.

3.1.1 G.711

Among all the available CODECs, G.711 is one of the most common and basic CODEC which has been used by number of manufacturers. It uses Pulse Code Modulation (PCM) "technique of voice frequencies at the rate of 64 kbps which covers both encoding methods "A-law" and " μ -law". A-law and μ -law are compounding schemes which facilitate linear coding to use more dynamics to the 8 bit samples. The voice signal is sample into 13 bit signed linear audio sample sampled at a sample rate, which is then compounded to 8 bit using a logarithmic scale for transmission over a 64 Kbps data channel of 8khz at the receiving end the data is then converted back to linear scale (13 bit) and played back. North America and Japan are mostly use μ -law whereas Europe and the rest of the world use A-law especially for the international routes. G.711 is a non-compressing CODEC, requires low computation complexity and provides very good voice quality with negligible delay. However, it consumes 64 kbps per direction, which is high compared to other CODEC (Mehdi, 2009).

3.1.2 G.723

There are two types of G.723 CODECs available in the market, one with the bit rate of 5.3 kbps and the other is 6.3 kbps, also denoted as G.723r53 and G.723r63, respectively. The higher bit rate corresponds to better quality whereas lower bit rate provides fair quality but provides system architecture with additional flexibility to use it for a bit rate (Mehdi, 2009).

3.1.3 G.729

The G.729 CODEC samples the filtered voice band at 8 kHz with a 16 bit resolution, it uses additional compressing algorithm to deliver a stream of 8 kbps. This special CODEC optimizes the bandwidth used for each connection. It normally requires a high computation complexity which introduces a relatively low delay. G.729 CODEC is transmitted using Real Time Protocol (RTP) over User Datagram Protocol (UDP) over Internet Protocol (IP) and the overhead introduced in VoIP communication links by the RTP/UDP/IP header which is quite high (Mehdi, 2009).

The following Table 3.1 summarizes common CODEC characteristics for the smallest packet duration, referred to as basic rate, and quality.

Table 3.1 CODEC performances comparison chart (Iqbal & Cheema, 2009)

Codec	Bit Rate	Method	Algorithm Delay	Quality (MOS)
G.711	64	A-law or μ -law	0.125ms	4.0
G.723r53	5.3	ACELP	37.5ms	3.6
G.723r63	6.3	MP-MLQ	37.5ms	3.9
G.729	8	CS-ACELP	15.0ms	3.9

3.2 VoIP Protocols

The CODEC needs a protocol to transport this data (coded speech) from one place to another which shows that the protocols are as important as the CODECs for the complete communication.

Protocols are set of rules or procedures that are either way used by endpoints when they communicate in a network. In Internet telephony data is transmitted in the form of Datagram. Every Datagram has a source address, destination address and sequence number. Each datagram of file/message is independently routed across the network and datagram are reassembled at the receiving end. The internet was designed to deliver the datagrams reliably without considering delays. Internet data transmissions are composed of several layers. The network layer consists of the IP which establishes a connection between two computers. The Internet Protocol (IP) is provided for routing datagrams between any two nodes with checking for corruption and loss. The transport layer provides the rules required for sending the data and the application layer determines how the data will be processed once it arrives at its destination.

Most data travelling over the Internet uses the Transmission Control Protocol (TCP) for the transport layer because it guarantees data delivery and integrity. TCP is provided for re-transmission of lost data and acknowledgements have also been sent back. Retries for re-transmission of data will be take some time, so then TCP can take much longer time. Thus TCP is highly unsatisfactory for fixed data transmission. VoIP does not need the kind of delivery guarantee which TCP provides, so IP network in VoIP transmissions can use an alternative faster transport layer protocol, user datagram protocol (UDP). UDP does not re-transmit the lost data and there are no acknowledgements also in case of UDP. However, in TCP if there are more number of hops, the acknowledgement takes longer time, but in UDP no such acknowledgement. Thus, UDP competes more effectively than TCP in a congested IP network for available bandwidth. Because of these reasons, VoIP generally uses UDP. Once VoIP data arrives at its destination, the application layer

interprets it and presents it to the user. In this chapter, the application layer protocols are examined.

In the application layer, Voice over IP (VoIP) uses signaling protocols (H.323, SIP, and MGCP) for establishing connections between endpoints and also, it uses media protocols (RTP, RTCP and RTSP) for dealing with the real time data such as audio or video. The most commonly used application layers for VoIP are SIP and RTP.

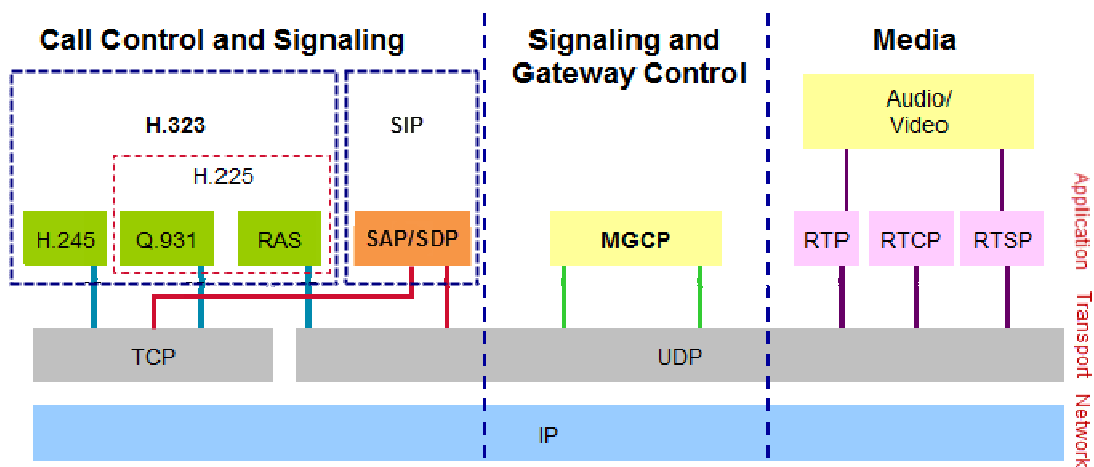


Figure 3.1 Pictorial overview for VoIP protocols (Minoli, 2006).

3.2.1 Signaling Protocols

Once a user dials a telephone number, signaling is required to determine the status of the called party (available or busy) and to establish the call. Call signaling is used in Voice over IP (VoIP) systems to establish connections between endpoints, or between an endpoint and a gatekeeper. VoIP signaling protocols are divided into two categories:

- **Session Control Protocols:** Session Control Protocols are responsible for the establishment, preservation and tearing down of call sessions. They are also responsible for the negotiation of session parameters such as codecs, tones, bandwidth capabilities, etc. The main Session Control Protocols in the IP network are H.323 and SIP.

- **Media Control Protocols:** Media Control Protocols are responsible for the creation and tearing down of media connections. They are used to open and close media pin-holes on VoIP gateways and to process notifications coming from those gateways. The Media Gateways are the VoIP components that transport media between the IP and PSTN networks. They are controlled by an entity that is called Media Gateway Controller. The latter uses a Media Control Protocol to control Media flows on the Gateway. The two main Media Control Protocols are MGCP and Megaco (H.248).

3.2.1.1 H.323

H.323 protocol specifies the components, protocols, and processes that provide multimedia communication services, real-time audio, video, and data communications over packet-based networks including the Internet. H.323 is part of a family of ITU-T recommendations called H.32x that provides multimedia communication services over a variety of networks. H.323 can be applied in a variety of mechanisms, such as audio only (IP telephony), audio and video (video telephony), audio and data, and audio, video and data. H.323 can also be applied to multipoint-multimedia communications.

3.2.1.1.1 H.323 Components. The H.323 standard specifies the following components. These are Terminals, Gateways (GW), Gatekeepers (GK), Multipoint Control Units (MCU), Multipoint Controller (MC), and Multipoint Processors (MP).

- **Terminal:** An H.323 terminal is an endpoint on the network which provides real-time, two-way communications with another H.323 terminal, GW, or MCU. This communication consists of control, indications, audio, moving color video pictures, and/or data between the two terminals. A terminal may provide speech only, speech and data, speech and video, or speech, data, and video (Kashihara, 2011).

- Gateway: The GW is a H.323 entity on the network which allows intercommunication between IP networks and legacy circuit-switched networks, such as ISDN and PSTN. They provide signaling mapping as well as transcoding facilities (Kashihara, 2011).
- Gatekeeper: The GK is a H.323 entity on the network which performs the role of the central manager of VoIP services to the endpoints. This entity provides address translation and controls access to the network for H.323 terminals, GWs, and MCUs. The GK may also provide other services to the terminals, GWs, and MCUs such as bandwidth management and locating GWs (Kashihara, 2011).
- MCU: The MCU is an H.323 entity on the network which provides the capability for three or more terminals and GW to participate in a multipoint conference. It may also connect two terminals in a point-to-point conference which may later develop into a multipoint conference. The MCU consists of two parts, a mandatory MC, and an optional MP. In the simplest case, an MCU may consist only of an MC with no MPs (Kashihara, 2011).
- MC: The MC is an H.323 entity on the network which controls three or more terminals participating in a multipoint conference. It may also connect two terminals in a point-to-point conference which may later develop into a multipoint conference. The MC provides the capability of negotiation with all terminals to achieve common levels of communications. It may also control conference resources such as who is multicasting video. The MC does not perform mixing or switching of audio, video, and data (Kashihara, 2011).
- MP: The MP is an H.323 entity on the network which provides for the centralized processing of audio, video and/or data streams in a multipoint conference. The MP provides for the mixing, switching, or other processing of media streams under the control of the MC. The MP may process a single

media stream or multiple media streams depending on the type of conference supported (Kashihara, 2011).

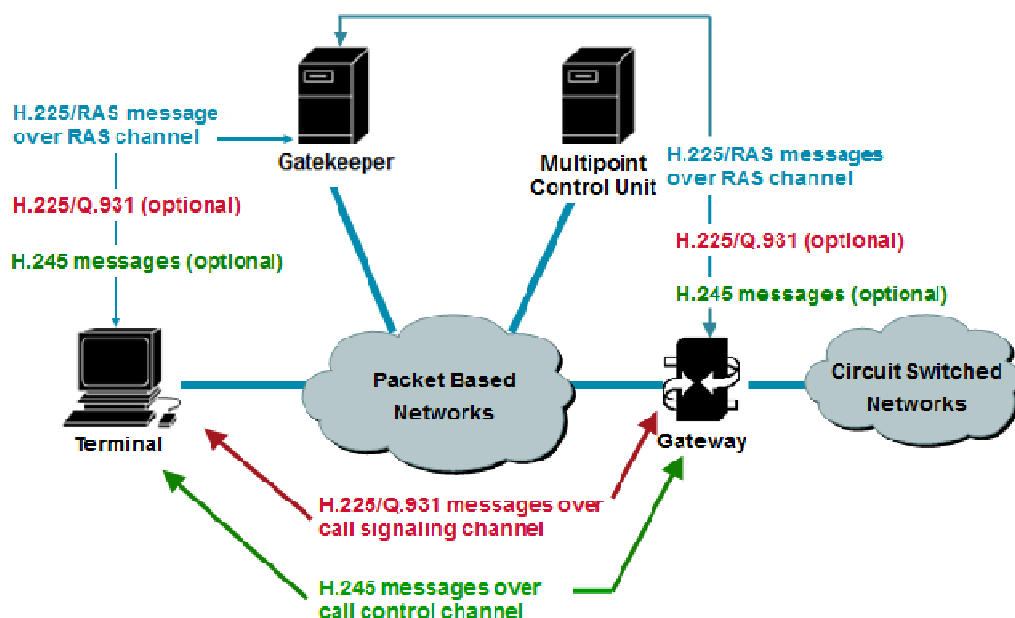


Figure 3.2 H.323 components and signaling (Minoli, 2006).

3.2.1.1.2 H.323 Protocols. H.323 is an umbrella recommendation which depends on several other standards and recommendations to enable real-time multimedia communications. The main ones are:

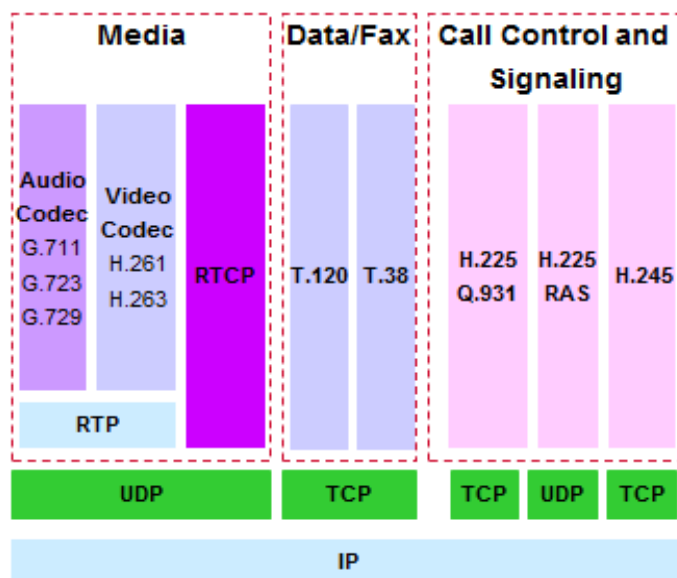


Figure 3.3 H.323 is an “Umbrella” specification (Minoli, 2006).

- **Audio CODEC:** Audio Codec encodes the audio signal from a microphone for transmission on the transmitting H.323 terminal and decodes the received audio code that is sent to the speaker on the receiving H.323 terminal. Because audio is the minimum service provided by the H.323 standard, all H.323 terminals must have at least one audio CODEC support, as specified in the ITU G.711 recommendation (audio coding at 64 kbps). Additional audio CODEC recommendations such as G.722 (64, 56, and 48 kbps), G.723.1 (5.3 and 6.3 kbps), G.728 (16 kbps), and G.729 (8 kbps) may also be supported (Tong, 2005).
- **Video CODEC:** Video Codec encodes video from a camera for transmission on the transmitting H.323 terminal and decodes the received video code that is sent to the video display on the receiving H.323 terminal. Because H.323 specifies support of video as optional, the support of video CODECs is optional as well. However, any H.323 terminal providing video communications must support video encoding and decoding as specified in the ITU H.261 recommendation (Tong, 2005).
- **H.225 Registration, Admission, and Status (RAS):** is the protocol used between endpoints (terminals and gateways) and gatekeepers to perform registration, admission control, bandwidth changes, status, and disengage procedures between endpoints and gatekeepers. A RAS channel exchanges RAS messages. This signaling channel is opened between an endpoint and a gatekeeper prior to the establishment of any other channels (Tong, 2005).
- **H.225 call signaling:** It establishes a connection between two H.323 endpoints. This is achieved by exchanging H.225 protocol messages on the call- signaling channel. The call- signaling channel is opened between two H.323 endpoints or between an endpoint and the gatekeeper (Tong, 2005).
- **H.245 control signaling:** It exchanges end-to-end control messages governing the operation of the H.323 endpoint. These control messages carry information

related to capability exchange, opening and closing of logical channels used to carry media streams, flow-control messages, general commands and indications (Tong, 2005).

3.2.1.1.3 H.323 Call Scenarios. Figure 3.4 shows a typical call flow for H.323 call setup between two endpoints registered to a gatekeeper.

- Both endpoints have previously registered with the gatekeeper.
- Terminal A initiates the call to the gatekeeper. (RAS messages are exchanged).
- The gatekeeper provides information for Terminal A to contact Terminal B.
- Terminal A sends a SETUP message to Terminal B.
- Terminal B responds with a Call Proceeding message and also contacts the gatekeeper for permission.
- Terminal B sends an Alerting and Connect message.
- Terminal B and A exchange H.245 messages to determine master slave, terminal capabilities, and open logical channels.
- The two terminals establish RTP media paths.

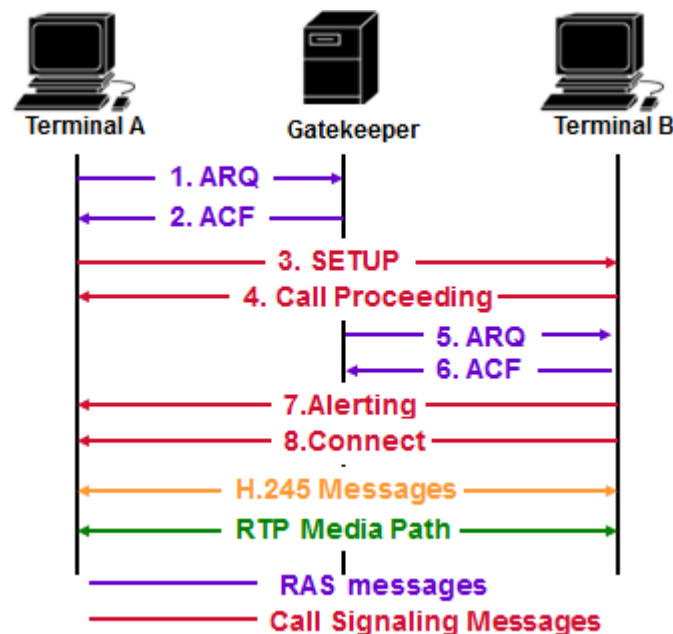


Figure 3.4 Call setup with H.323 (Minoli, 2006).

3.2.1.2 SIP (*Session Initiation Protocol*)

SIP was developed by IETF in reaction to the ITU-T H.323 recommendation. The IETF believed that H.323 was inadequate for evolving IP telephony, because its command structure is complex and its architecture is centralized and monolithic. SIP is an application layer control protocol that can establish, modify, and terminate multimedia sessions or calls (Kashihara, 2011).

The architecture of SIP is similar to that of HTTP (client-server protocol). Requests are generated by the client and sent to the server. The server processes the requests and then sends a response to the client. A request and the responses for that request make a transaction. SIP has INVITE and ACK messages which define the process of opening a reliable channel over which call control messages may be passed. SIP makes minimal assumptions about the underlying transport protocol. This protocol itself provides reliability and does not depend on TCP for reliability. SIP depends on the Session Description Protocol (SDP) for carrying out the negotiation for codec identification. SIP supports session descriptions that allow participants to agree on a set of compatible media types. It also supports user mobility by proxying and redirecting requests to the user's current location. The services that SIP provides include:

- User Location: determination of the end system to be used for communication
- Call Setup: ringing and establishing call parameters at both called and calling party
- User Availability: determination of the willingness of the called party to engage in communications
- User Capabilities: determination of the media and media parameters to be used
- Call handling: the transfer and termination of calls (Tong, 2005)

3.2.1.2.1 SIP Components. A system using SIP can be viewed as consisting of components defined on two dimensions: client/server and individual network elements. RFC3261 defines client and server as follows:

- Client: A client is any network element that sends SIP requests and receives SIP responses. Clients may or may not interact directly with a human user. User agent clients and proxies are clients (Stallings, 2003).
- Server: A server is a network element that receives requests in order to service them and sends back responses to those requests. Examples of servers are proxies, user agent servers, redirect servers, and registrars (Stallings, 2003).

The individual elements of a standard SIP configuration include the following:

- User Agents: It is an application that interacts with the user and contains both a User Agent Client (UAC) and User Agent Server (UAS). A user agent client initiates SIP requests, and a user agent server receives SIP requests and returns responses on user behalf (Kashihara, 2011).
- Registrar Server: It is a SIP server that accepts only registration requests issued by user agents for the purpose of updating a location database with the contact information of the user specified in the request (Kashihara, 2011).
- Proxy Server: It is an intermediary entity that acts both as a server to user agents by forwarding SIP requests and acts as a client to other SIP servers by submitting the forwarded requests to them on behalf of user agents or proxy servers (Kashihara, 2011).
- Redirect Server: It is a SIP server that helps to locate UAs by providing alternative locations where the user can be reachable, i.e., provides address mapping services. It responds to a SIP request destined to an address with a list

of new addresses. A redirect server does not accept calls, does not forward requests, and does not initiate any of its own (Kashihara, 2011).

- **Location Service:** A location service is used by a SIP redirect or proxy server to obtain information about a callee's possible location(s). For this purpose, the location service maintains a database of SIP-address/IP-address mappings (Stallings, 2003).

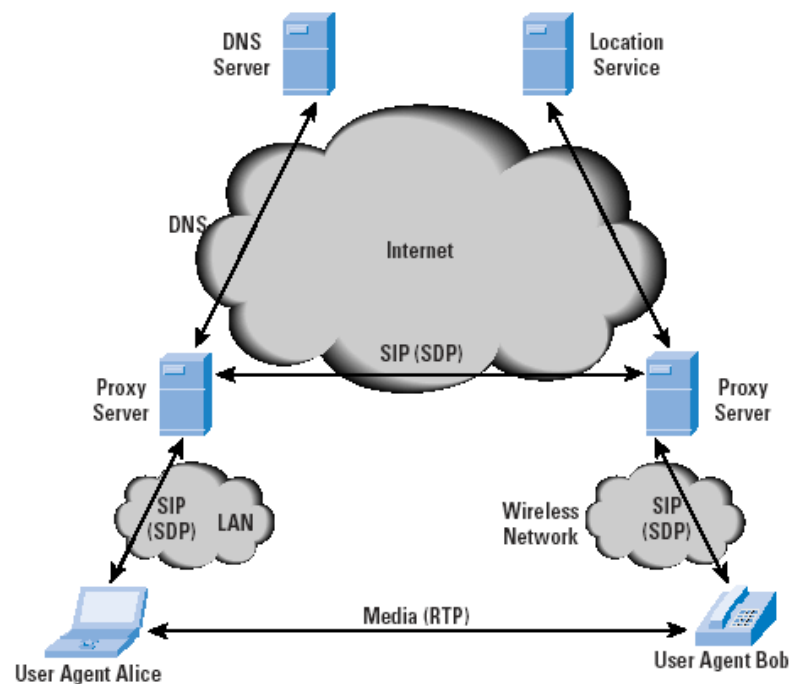


Figure 3.5 SIP components and protocols (Stallings, 2003).

Figure 3.5 shows how some of the SIP components relate to one another and the protocols that are employed. A user agent acting as a client (in this case UAC Alice) uses SIP to set up a session with a user agent that acts as a server (in this case UAS Bob). The session initiation dialogue uses SIP and involves one or more proxy servers to forward requests and responses between the two user agents. The user agents also make use of the SDP, which is used to describe the media session (Stallings, 2003).

The proxy servers may also act as redirect servers as needed. If redirection is done, a proxy server needs to consult the location service database, which may or

may not be collocated with a proxy server. The communication between the proxy server and the location service is beyond the scope of the SIP standard. The *Domain Name System* (DNS) is also an important part of SIP operation. Typically, a UAC makes a request using the domain name of the UAS, rather than an IP address. A proxy server needs to consult a DNS server to find a proxy server for the target domain (Stallings, 2003).

3.2.1.2.2 SIP Protocols. SIP often runs on top of the *User Datagram Protocol* (UDP) for performance reasons, and provides its own reliability mechanisms, but it may also use TCP. If a secure, encrypted transport mechanism is desired, SIP messages may alternatively be carried over the *Transport Layer Security* (TLS) protocol (Stallings, 2003).

Associated with SIP is the SDP, defined in RFC 2327. It describes the content of sessions, including telephony, internet radio and multimedia applications. SDP includes information about:

- Media streams: A session can include multiple streams of differing content. SDP currently defines audio, video, data, control, and application as stream types, similar to the MIME types used for Internet mail.
- Addresses: SDP indicates the destination addresses, which may be a multicast address, for a media stream.
- Ports: For each stream, the UDP port numbers for sending and receiving are specified.
- Payload types: For each media stream type in use (for example, telephony), the payload type indicates the media formats that can be used during the session.

- Start and stop times: These apply to broadcast sessions, for example, a television or radio program. The start, stop, and repeat times of the session are indicated.
- Originator: For broadcast sessions, the originator is specified, with contact information. This may be useful if a receiver encounters technical difficulties.

Although SDP provides the capability to describe multimedia content, it lacks the mechanisms by which two parties agree on the parameters to be used. RFC 3264 remedies this lack by defining a simple offer/answer model, by which two parties exchange SDP messages to reach agreement on the nature of the multimedia content to be transmitted. After this information is exchanged and acknowledged, all participants are aware of the participants' IP addresses, available transmission capacity, and media type. Then, data transmission begins, using an appropriate transport protocol. Typically, the RTP is used. Throughout the session, participants can make changes to session parameters, such as new media types or new parties to the session, using SIP messages (Stallings, 2003).

3.2.1.2.3 SIP Call Scenarios. SIP embarks on a four-step procedure to construct a VoIP call, from a signaling viewpoint. First, a caller locates the appropriate server, then sends a SIP request (usually “invite”). Typically, the request arrives at its destination, where the client accepts the call. Then the originating caller sends an acknowledgement back to the recipient. Likewise, the station that initiates the call also sends the acknowledgement. The detailed information about this procedure is explained in chapter three.

3.2.1.3 Comparison between SIP and H.323

H.323 and SIP are both competing for the dominance of IP telephony signaling. There is much debate in the industry as to which protocol is superior, H.323, SIP or perhaps another protocol that may be in the early stages of development. Currently,

there is no clear-cut winner. The main differences of SIP and H.323 are summarized in the table 3.2.

Table 3.2 Comparison between H.323 and SIP (Tong, 2005)

Area	H.323	SIP
Complexity	Complex protocol	Comparatively simpler
Encoding	Binary ASN.1 PSN encoding	Text-based UTF-8 encoding
Extensibility	Limited	Easy, not limited
Compatibility	Requires full backward compatibility	Does not require full backward compatibility
Scalability	Less scalable (state full, TCP)	More scalable (stateless, UDP)
Transport	TCP only	TCP, UDP or other
Conferencing	MCU required	Using IP multicast
Services	Provide richer set of functionality	Simple set of functionality
Loop detection	State full (difficult)	Stateless (comparatively easy)
Addressing	E.164 scheme, H.323 ID alias, ... (more flexible)	SIP URLs
Mobility	More limited (does not support forking proxy)	More flexible and rapid (support forking proxy)
Conference control	Supported	Not supported

- **Complexity:** If we compare the protocols in the aspect of complexity, H.323 is the most complex of the two protocols. H.323 defines hundreds of elements, while SIP has only 37 headers, each with a small number of values and parameters. H.323 uses a binary representation for its messages, which are based on Abstract Syntax Notation One (ASN.1) and the packed encoding rules

(PER). ASN.1 generally requires special code-generators to parse. SIP uses a simple format for commands and messages, the text format similar to HTTP and RTSP. These are text strings that are easy to decode, and hence, easy to debug. The entire set of messages is also much smaller than in H.323. Another advantage of SIP is that it uses a single request that contains all necessary information, while many of the H.323 services require interaction between the several protocol components that are included in the standard (Tong, 2005).

- **Compatibility:** H323 is a strict protocol; it requires full backward compatibility. That means the later version of H323 must be compatible with the earlier version. In a H323 system, a Cisco gateway must co-operates with a terminal produced by Lucent because of standard implementations. Otherwise, SIP is an open protocol and easy to extend. It does not require full backward compatibility; it means the later version does not have to support all the capability of previous versions. SIP devices can be easily compatible to systems of other producers just by exchanging information about their capabilities, such as encoding methods or the messages they have, and co-operate only on the common capability (Tong, 2005).
- **Scalability:** It is also important as the use of Internet and its services tend to grow. At below the protocols are compared in different levels:
 1. **Large Numbers of Domains:** As H.323 was originally meant to be used on a single LAN, it has some problems with the scalability even though the newest version defines the concept of zones, and defines procedures for user location across zones for email names. It provides no easy way to perform loop detection in complex multi-domain searches, it can be done state fully by storing messages but this is not scalable. SIP, however, uses a loop detection method by checking the history of the message in the via header fields, which can be performed in a stateless manner.
 2. **Server Processing:** Both H323 gateways, gatekeepers and SIP servers, gateways will be required to handle calls from a multitude of users. A SIP

transaction through several servers and gateways can be either state full or stateless. This means that large, backbone servers that handle a lot of traffic can be stateless to reduce the memory requirements. This is combined with the ability of using UDP, as UDP does not require any connection state. H.323, on the other, requires its gatekeepers to be state full. Furthermore, the connections are TCP based, which means that a gatekeeper must hold its connections throughout a call.

- **Mobility:** The service of personal mobility is also supported by both protocols, but H.323's support for this is more limited. SIP can both redirect and proxy incoming requests to a number of locations using any arbitrary URL. Information about language spoken, business or home, mobile phone or fixed, and a list of callee priorities, can be conveyed for each location. SIP also supports, multi-hop "searches" for a user. This means that the servers can proxy the request to one or more additional servers in search of the callee. A SIP server can also proxy the request to multiple servers in parallel, called forking proxy, which makes the search operation more rapid. H.323 can redirect a caller to try several other addresses. Here it is neither possible to express preferences, nor can the caller express preferences in the original invitation. H.323 was not designed for wide area operation, it does support call forwarding, but as mentioned before it has no mechanism for loop detection H.323 does not allow a gatekeeper to proxy a request to multiple servers either (Tong, 2005).
- **Services:** Roughly SIP and H.323 provides the same services, even if new services always are added. In addition to call control services, both SIP and H.323 provide capabilities exchange services. In this regard, H.323 provides a much richer set of functionality. Terminals can express their ability to perform various encodings and decodings based on parameters of the codec, and based on which other codecs are in use. SIP only uses basic receiver capability indication. This means that SIP sends a list of the encodings supported and it is for the other side to choose any subset of these (Tong, 2005).

3.2.1.4 MGCP (Media Gateway Control Protocol)

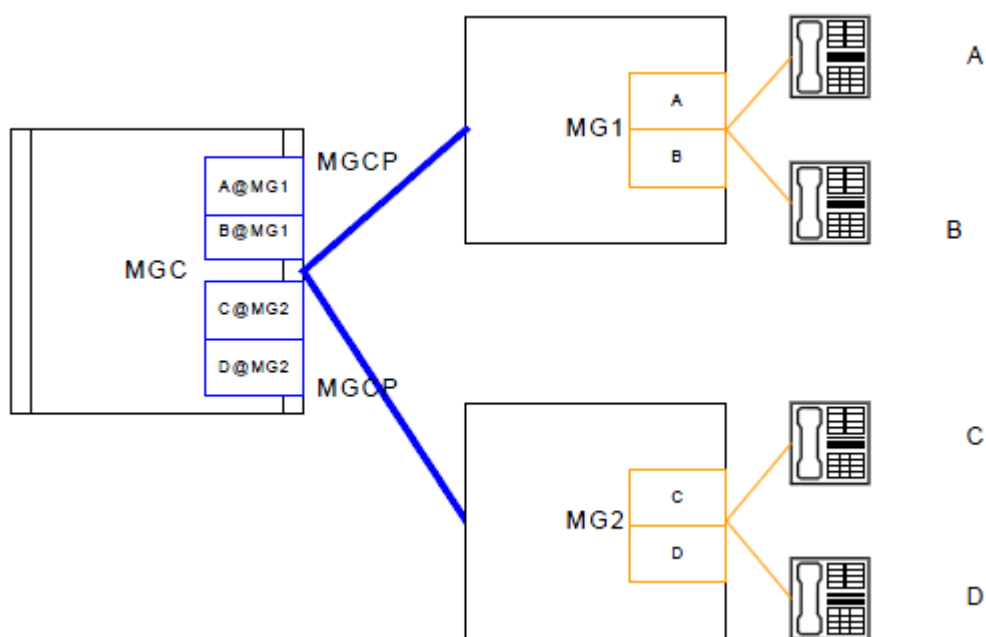


Figure 3.6 MGCP endpoints and connections

This protocol was the predecessor to ‘Megaco’ and still holds sway with a number of carriers and other VoIP users. It is a protocol that defines communication between call control elements (Call Agents) and telephony gateways. Call Agents are also known as Media Gateway Controllers. It is a control protocol, allowing a central coordinator to monitor events in IP phones and gateways and instructs them to send media to specific addresses. It resulted from the merger of the Simple Gateway Control Protocol and Internet Protocol Device Control. The call control intelligence is located outside the gateways and handled by external call control elements, the Call Agent. MGCP assumes that these call control elements or Call Agents will synchronize with each other to send coherent commands to the gateways under their controls. It is a master/slave protocol, where the gateways are expected to execute commands sent by the Call Agents. It has introduced the concepts of connections and endpoints for establishing voice paths between two participants, and the concepts of events and signals for establishing and tearing down calls. Since the main emphasis of MGCP is simplicity and reliability and it allows programming difficulties to be concentrated in Call Agents, so it will enable service providers to develop reliable and cheap local access systems.

3.2.1.5 Megaco/H.248

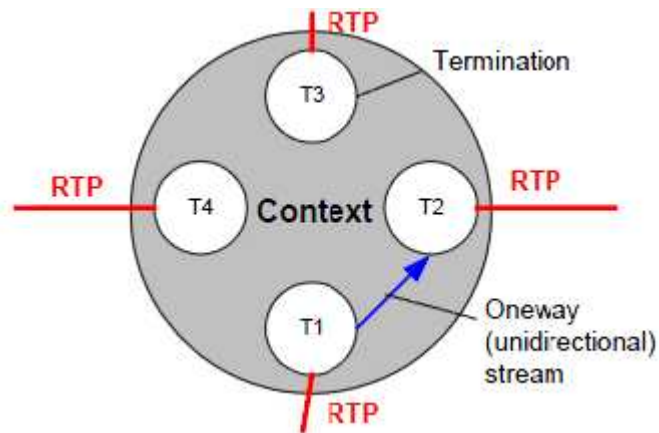


Figure 3.7 Megaco/H.248 concepts

Megaco is a call-control protocol that communicates between a gateway controller and a gateway. It evolved from and replaces SGCP (simple gateway control protocol) and MGCP (media gateway control protocol). Megaco addresses the relationship between a media gateway (MG) and a media gateway controller (MGC). An MGC is sometimes called a 'softswitch' or 'call agent'. Both Megaco and MGCP are relatively low-level devices that instruct MGs to connect streams coming from outside the cell or packet data network onto a packet or cell stream governed by RTP.

3.2.1.6 Comparison between MGCP and Megaco/H.248

MEGACO offers the following key enhancements over MGCP:

- Supports multimedia and multipoint conferencing-enhanced services
- Improved syntax for more efficient semantic message processing
- TCP and UDP transport options
- Allows text or binary encoding, formalized extension process for enhanced functionality
- Formalized extension process for enhanced functionality

Table 3.3 Main differences between Megaco/MGCP

Megaco /H.248	MGCP
A call is represented by terminations within a call context	A call is represented by endpoints within connections
Call types include any combination of multimedia and conferencing	Call types include point-to-point and multipoint
Syntax is text binary	Syntax is text
Transport layer is TCP or UDP	Transport layer is UDP
Defined by the IETF and ITU	Defined by Cisco and circulated in IETF

H.248 has the same architecture as MGCP. The commands are similar, but the main difference is that H.248 commands apply to terminations relative to a context rather than to individual connections, as is the case with MGCP. Connections are achieved by placing two or more terminations into a common context. It is the concept of a context that facilitates support of multimedia and conferencing calls. The context can be viewed as a mixing bridge that supports multiple media streams for enhanced multimedia services (Sulkin, 2002).

3.2.2 Real Time Protocols

The Internet carries all types of traffic. Each type has different characteristics and requirements. For example, a file transfer application requires that some quantity of data is transferred in an acceptable amount of time, while Internet telephony requires that most packets get to the receiver in less than 0.3 seconds. If enough bandwidth is available, best-effort service fulfills all of these requirements. When resources are scarce, however, real-time traffic will suffer from the congestion (Liu, 1998).

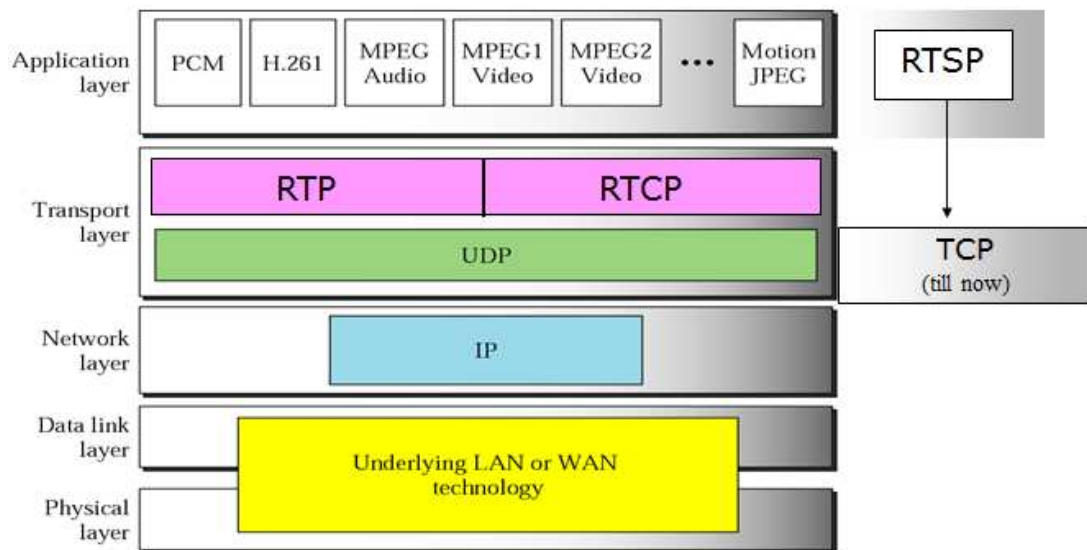


Figure 3.8 Protocol stack for multimedia services (Hetawal, 2005).

The solution for multimedia over IP is to classify all traffic, allocate priority for different applications and make reservations. The Integrated Services working group in the IETF (Internet Engineering Task Force) developed an enhanced Internet service model called Integrated Services that includes best-effort service and real-time service, see RFC 1633. The real-time service will enable IP networks to provide quality of service to multimedia applications. Resource Reservation Protocol (RSVP), together with Real-time Transport Protocol (RTP), Real-Time Control Protocol (RTCP), Real-Time Streaming Protocol (RTSP), provides a working foundation for real-time services. Integrated Services allows applications to configure and manage a single infrastructure for multimedia applications and traditional applications. It is a comprehensive approach to provide applications with the type of service they need and in the quality they choose (Liu, 1998).

3.2.2.1 RTP (*Real Time Transport Protocol*)

Real-Time Transport Protocol (RTP) is the Internet protocol which transmits real-time data such as audio and video. RTP does not exclusively guarantee real-time delivery of data, but it does provide mechanisms for the sending and receiving applications to support streaming data.

As VoIP doesn't use TCP (Transmission Control Protocol), RTP runs on top of the User Datagram protocol (UDP) instead. VoIP uses UDP as the transport layer. The UDP protocol provides only a direct method of sending and receiving data over an IP network and offers very few error recovery services. UDP has no mechanisms in place to notify the application of any loss in transmission whilst delivering packets of data; it also sends data unordered with no guarantees of the data being presented in the receiving application. All re-ordering of data into the correct format, which it was sent, is handled by the RTP.

When transmitting the streams of data, the protocol needs to handle the following conditions in the network:

- The network can de-sequence packets
- Some packets can be lost
- Jitter is introduced (jitter is a variance of packet inter-arrival time).

Out of these three, RTP aims to solve only two issues, packet de-sequencing and jitter (using sequence numbers and timestamps). When it comes to packet loss, the protocol prefers "real-timeless" to reliability. If some packets get lost, they get lost, it's more important to transmit the stream in real time. Because of this, RTP works on top of UDP. TCP is not suitable for real-time protocols because of its retransmission scheme.

In the Figure 3.9, a simplified RTP packet structure is shown. The most important fields of this packet are payload type, sequence number, timestamp, synchronization source and contributing source.

Payload type for the data carried in the packet. The PT field is 7 bit long, so it allows values between 0 and 127. There are several static values defined, for example "0" represents G.711 u-Law, "8" represents G.711 A-Law, and "18" stands for G.729. The interval between 96 and 127 is reserved for dynamic payload types.

These dynamic payload types need to be negotiated by whatever signaling protocol is used to establish the VoIP call (e.g. SIP or H.323).

The sequence number starts at a random value and is incremented with each RTP packet sent. This helps to identify packets received out of sequence. Similar to the sequence number, the timestamp is initialized with a random value. The clock frequency depends on the payload type. With the most usual narrow-band audio, the frequency is 8000 Hz and the timestamp is the tick count when the first audio sample in the payload was sampled.

Synchronization source (SSRC) is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier. In a special situation, the stream can be produced by a mixer from several streams. The IDs of the contributing sources can be listed in the CSRC fields and the field CC gives the number of contributing sources. However, this is not used very often in practice.

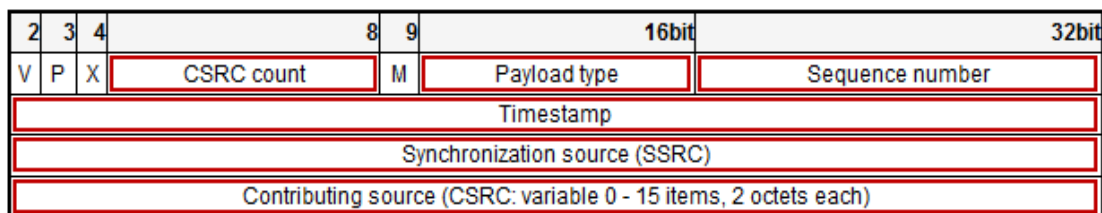


Figure 3.9 The RTP packet header.

In the most typical situation (no CSRC fields, no header extension), the RTP header consists of 12 bytes. In VoIP, voice packets are inserted into data packets using RTP, which in turn are inside UDP packets. Once VoIP data arrives, the application layer interprets it and the data is presented to the user.

3.2.2.2 RTCP (Real Time Control Protocol)

RTCP accompanies RTP and is used to transmit control information about the RTP session. RTCP packets are sent only from time to time since there is a

recommendation that the RTCP traffic should consume less than 5 percent of the session bandwidth.

The most important content types carried in RTCP packets include information about call participants (for example, name and e-mail address) and statistics about the quality of the transmission (for example inter-arrival jitter and the number of lost packets). The report sent by a participant who both sends and receives data is called a sender report (SR), while reports sent by participants who only receive RTP streams are called receiver reports (RR).

There is a rule that RTP should use an even UDP port number (e.g. 5000) and the related RTCP should use the next odd port (e.g. 5001).

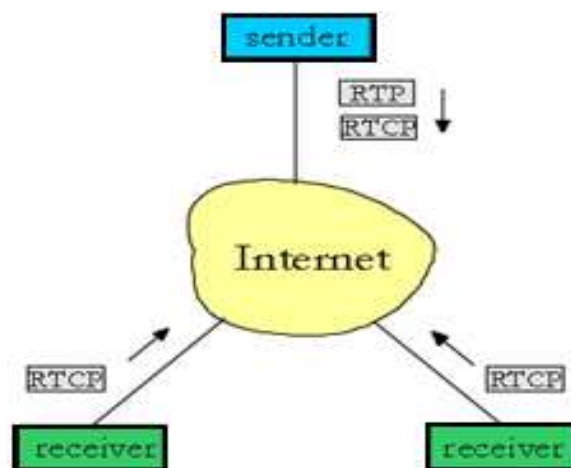


Figure 3.10 The real time protocols (Hetawall, 2005).

RTCP performs four functions. The first one is providing feedback on the quality of the data distribution. This is an integral part of the RTP's role as a transport protocol and is related to the flow and congestion control functions of other transport protocols. The second one is carrying a persistent transport-level identifier for an RTP source called the canonical name or CNAME. Since the SSRC identifier may change if a conflict is discovered or a program is restarted, receivers require the CNAME to keep track of each participant. Receivers may also require the CNAME to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video. The first two functions require

that all participants send RTCP packets, therefore the rate must be controlled in order for RTP to scale up to a large number of participants. By having each participant send its control packets to all the others, each can independently observe the number of participants. This number is used to calculate the rate at which the packets are sent. An optional function is to convey minimal session control information, for example participant identification to be displayed in the user interface. This is most likely to be useful in "loosely controlled" sessions where participants enter and leave without membership control or parameter negotiation (Schulzrinne & Casner, 2003).

The first three functions should be used in all environments, but particularly in the IP multicast environment. RTP application designers should avoid mechanisms that can only work in unicast mode and will not scale to larger numbers. Transmission of RTCP may be controlled separately for senders and receivers for cases such as unidirectional links where feedback from receivers is not possible.

3.2.2.3 RTSP (Real Time Streaming Protocol)

RTSP, the Real Time Streaming Protocol, is a client-server protocol that provides control over the delivery of real-time media streams. It provides "VCR-style" remote control functionality for audio and video streams, like pause, fast forward, reverse, and absolute positioning. It provides the means for choosing delivery channels (such as UDP, multicast UDP and TCP), and delivery mechanisms based upon RTP. RTSP establishes and controls streams of continuous audio and video media between the media servers and the clients. A media server provides playback or recording services for the media streams while a client requests continuous media data from the media server. RTSP acts as the "network remote control" between the server and the client (Arora, 1999).

It supports the following operations:

- Retrieval of media from media server: The client can request a presentation description, and ask the server to setup a session to send the requested data.

The server can either multicast the presentation or send it to the client using unicast.

- Invitation of a media server to a conference: The media server can be invited to the conference to play back media or to record a presentation.
- Addition of media to an existing presentation: The server or the client can notify each other about any additional media that has become available.

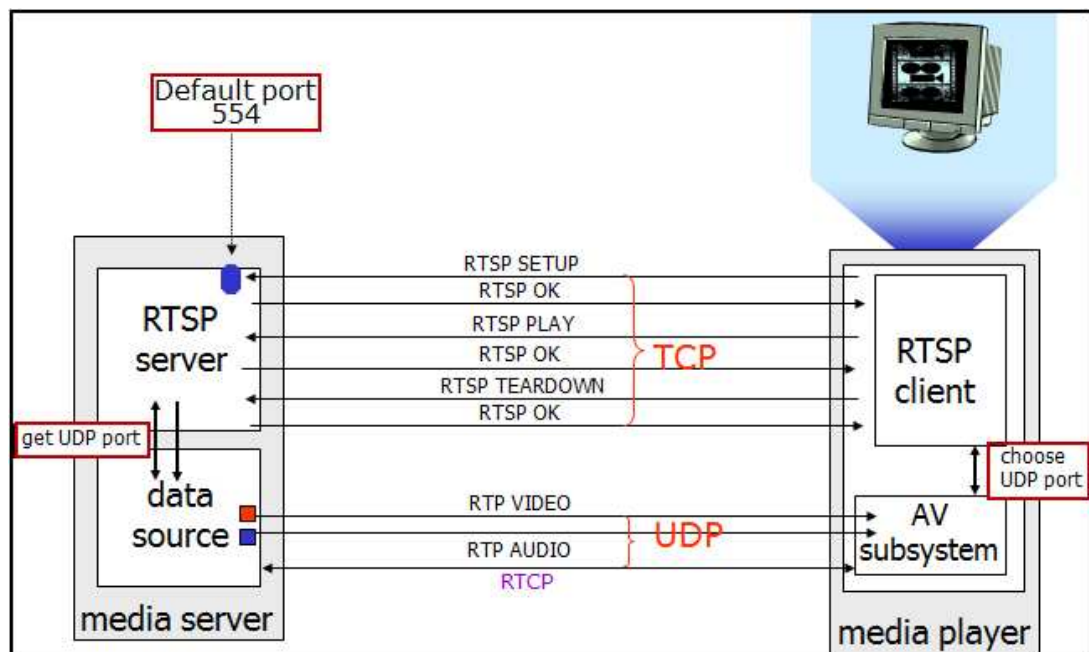


Figure 3.11 The RTSP session (Hetawall, 2005).

Features of RTSP include:

- RTSP is an application level protocol with syntax and operations similar to HTTP, but works for audio and video. It uses URLs like those in HTTP.
- An RTSP server needs to maintain states, using SETUP, TEARDOWN and other methods.
- Unlike HTTP, in RTSP both servers and clients can issue requests.
- RTSP is implemented on multiple operating system platforms and it allows interoperability between clients and servers from different manufacturers.

3.2.2.4 RSVP (Resource Reservation Protocol)

A host uses RSVP to request a specific Quality of Service (QoS) from the network, on behalf of an application data stream. RSVP carries the request through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream (Berson, 1999).

To make a resource reservation at a node, the RSVP daemon communicates with two local decision modules, admission control and policy control. Admission control determines whether the node has sufficient available resources to supply the requested QoS. Policy control determines whether the user has administrative permission to make the reservation. If either check fails, the RSVP program returns an error notification to the application process that originated the request. If both checks succeed, the RSVP daemon sets parameters in a packet classifier and packet scheduler to obtain the desired QoS. The packet classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream.

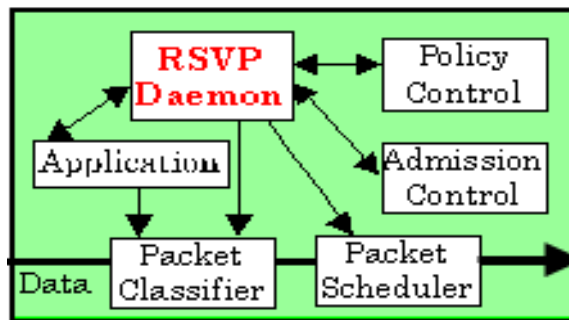


Figure 3.12 RSVP modules (Berson, 1999).

A primary feature of RSVP is its scalability. RSVP scales to very large multicast groups because it uses receiver-oriented reservation requests that merge as they progress up the multicast tree. The reservation for a single receiver does not need to travel to the source of a multicast tree; rather it travels only until it reaches a reserved branch of the tree. The reservation request merges as it travels up the multicast tree.

While the RSVP protocol is designed specifically for multicast applications, it may also make unicast reservations.

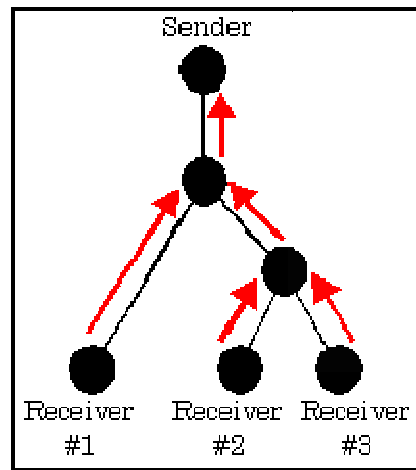


Figure 3.13 RSVP multicast tree
(Berson, 1999).

RSVP is also designed to utilize the robustness of current Internet routing algorithms. RSVP does not perform its own routing; instead it uses underlying routing protocols to determine where it should carry reservation requests. As routing changes paths to adapt to topology changes, RSVP adapts its reservation to the new paths wherever reservations are in place. This modularity does not rule out RSVP from using other routing services. Current research within the RSVP project is focusing on designing RSVP to use routing services that provide alternate paths and fixed paths.

CHAPTER FOUR

SIP OPERATIONS

4.1 Introduction

In this chapter, the examples of Session Initiation Protocol (SIP) call flows are examined. Elements in these call flows include SIP User Agents and Clients, SIP Proxy and Redirect Servers. Scenarios include SIP registration and SIP session establishment. Call flow diagrams and message details are shown.

A resource within a SIP configuration is identified by a URI. Examples of communications resources include the following:

- A user of an online service
- An appearance on a multiline phone
- A mailbox on a messaging system
- A telephone number at a gateway service
- A group (such as “sales” or “help desk”) in an organization

SIP URIs has a format based on e-mail address formats, namely **user@domain**. There are two common schemes. An ordinary SIP URI is of the form: **sip:bob@biloxi.com**. The URI may also include a password, port number, and related parameters. If secure transmission is required, “**sip:**” is replaced by “**sips:**”. In the latter case, SIP messages are transported over TLS (Stallings, 2003).

4.2 SIP Messages

SIP is a text-based protocol with syntax similar to that of HTTP. There are two different types of SIP messages, requests and responses. The format difference between the two types of messages is seen in the first line. The first line of a request has a method, defining the nature of the request and a Request-URI, indicating where the request should be sent. The first line of a response has a response code. All

messages include a header, consisting of a number of lines, each line beginning with a header label. A message can also contain a body such as an SDP media description.

For SIP requests, RFC 3261 defines the following methods:

- **REGISTER:** Used by a user agent to notify a SIP configuration of its current IP address and the URLs for which it would like to receive calls.

In the following figures (Figure 4.1 and Figure 4.2), the scenarios about SIP registration are shown. In the Figure 4.1 Bob sends a SIP REGISTER request to the SIP server. The request includes the user's contact list. This flow shows the use of HTTP Digest for authentication using TLS transport. TLS transport is used due to the lack of integrity protection in HTTP Digest and the danger of registration hijacking without it, as described in RFC 3261. The SIP server provides a challenge to Bob. Bob enters her/his valid user ID and password. Bob's SIP client encrypts the user information according to the challenge issued by the SIP server and sends the response to the SIP server. The SIP server validates the user's credentials. It registers the user in its contact database and returns a response (200 OK) to Bob's SIP client. The response includes the user's current contact list in Contact headers. The format of the authentication shown is HTTP digest. It is assumed that Bob has not previously registered with this Server (Johnston, Donovan & Sparks, 2003).

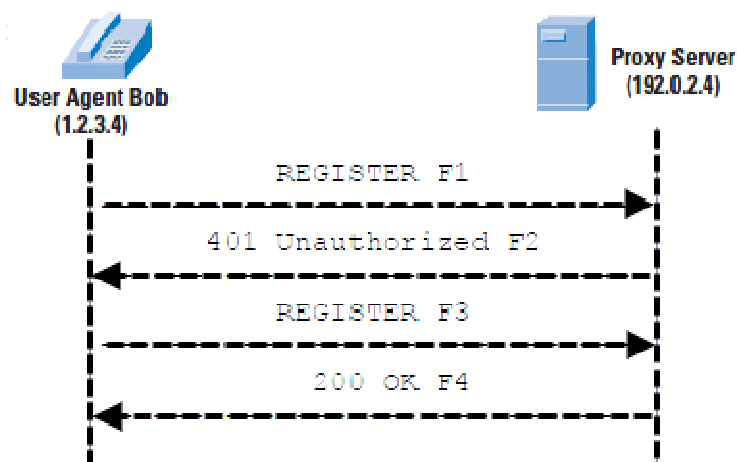


Figure 4.1 Successful new registration (Johnston, Donovan & Sparks, 2003).

In the Figure 4.2 Bob sends a SIP REGISTER request to the SIP Server. The SIP server provides a challenge to Bob. Bob enters her/his user ID and password. Bob's SIP client encrypts the user information according to the challenge issued by the SIP server and sends the response to the SIP server. The SIP server attempts to validate the user's credentials, but they are not valid (the user's password does not match the password established for the user's account). The server returns a response (401 Unauthorized) to Bob's SIP client.

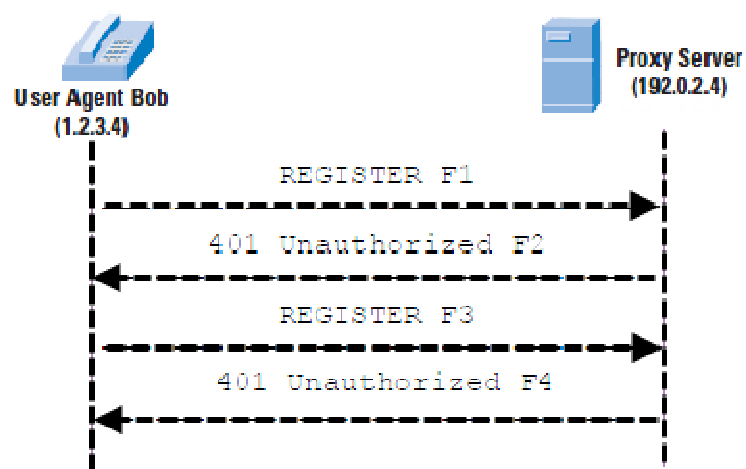


Figure 4.2 Unsuccessfull registration (Johnston, Donovan & Sparks, 2003).

- INVITE: Used to establish a media session between user agents
- ACK: Confirms reliable message exchanges
- CANCEL: Terminates a pending request, but does not undo a completed call
- BYE: Terminates a session between two users in a conference
- OPTIONS: Solicits information about the capabilities of the callee, but does not set up a call

For example, the header of message (1) in Figure 4.3 might look like the following:

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP 12.26.17.91:5060
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com;tag=1928301774>
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:alice@atlanta.com>
Content-Type: application/sdp
Content-Length: 142
```

The first line contains the method name (INVITE), a SIP URI, and the version number of SIP that is used. The lines that follow are a list of header fields. This example contains the minimum required set (Stallings, 2003).

The ‘Via’ headers show the path the request has taken in the SIP configuration (source and intervening proxies), and are used to route responses back along the same path. As the INVITE message leaves, there is only the header inserted by Alice. The line contains the IP address (12.26.17.91), port number (5060), and transport protocol (UDP) that Alice wants Bob to use in his response (Stallings, 2003).

The ‘Max-Forwards’ header limits the number of hops a request can make on the way to its destination. It consists of an integer that is decremented by one by each proxy that forwards the request. If the Max-Forwards value reaches 0 before the request reaches its destination, it is rejected with a 483 (Too Many Hops) error response (Stallings, 2003).

The ‘To’ header field contains a display name (Bob) and a SIP or SIPS URI (sip:bob@biloxi.com) toward which the request was originally directed. The ‘From’

header field also contains a display name (Alice) and a SIP or SIPS URI (sip:alice@atlanta.com) that indicate the originator of the request. This header field also has a tag parameter that contains a random string (1928301774) that was added to the URI by the UAC. It is used to identify the session.

The 'Call-ID' header field contains a globally unique identifier for this call, generated by the combination of a random string and the host name or IP address. The combination of the 'To' tag, 'From' tag, and 'Call-ID' completely defines a peer-to-peer SIP relationship between Alice and Bob and is referred to as a dialog.

The 'CSeq' or 'Command Sequence' header field contains an integer and a method name. The 'CSeq' number is initialized at the start of a call (314159 in this example), incremented for each new request within a dialog, and is a traditional sequence number. The 'CSeq' is used to distinguish a retransmission from a new request (Stallings, 2003).

The 'Contact' header field contains a SIP URI for direct communication between user agents. Whereas the 'Via' header field tells other elements where to send the response, the 'Contact' header field tells other elements where to send future requests for this dialog (Stallings, 2003).

The 'Content-Type' header field indicates the type of the message body. The 'Content-Length' header field gives the length in octets of the message body.

The SIP response types defined in RFC 3261 are in the following categories:

- Provisional (1xx): The request was received and is being processed.
- Success (2xx): The action was successfully received, understood, and accepted.
- Redirection (3xx): Further action needs to be taken in order to complete the request.
- Client Error (4xx): The request contains bad syntax or cannot be fulfilled at this server.

- Server Error (5xx): The server failed to fulfill an apparently valid request.
- Global Failure (6xx): The request cannot be fulfilled at any server.

For example, the header of message (13) in Figure 4.3 might look like the following:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
Via: SIP/2.0/UDP 12.26.17.91:5060
To: Bob <sip:bob@biloxi.com;tag=a6c85cf
From: Alice <sip:alice@atlanta.com;tag=1928301774
Call-ID: a84b4c76e66710@12.26.17.91
CSeq: 314159 INVITE
Contact: <sip:bob@biloxi.com>
Content-Type: application/sdp
Content-Length: 131
```

The first line contains the version number of SIP that is used and the response code and name. The lines that follow are a list of header fields. The ‘Via’, ‘To’, ‘From’, ‘Call-ID’, and ‘CSeq’ header fields are copied from the INVITE request. (There are three ‘Via’ header field values; one added by Alice’s SIP UAC, one added by the atlanta.com proxy, and one added by the biloxi.com proxy.) Bob’s SIP phone has added a tag parameter to the ‘To’ header field. This tag is incorporated by both endpoints into the dialog and is included in all future requests and responses in this call (Stallings, 2003).

4.3 SIP Session Establishment

Figure 4.3 shows a successful attempt by user Alice to establish a session with user Bob, whose URI is ‘bob@biloxi.com’. Alice’s UAC is configured to communicate with a proxy server (the outbound server) in its domain and begins by

sending an INVITE message to the proxy server that indicates its desire to invite Bob's UAS into a session (1); the server acknowledges the request (2). Although Bob's UAS is identified by its URI, the outbound proxy server needs to account for the possibility that Bob is not currently available or that Bob has moved. Accordingly, the outbound proxy server should forward the INVITE request to the proxy server that is responsible for the domain **biloxi.com**. The outbound proxy thus consults a local DNS server to obtain the IP address of the **biloxi.com** proxy server (3), by asking for the DNS SRV resource record that contains information on the proxy server for **biloxi.com** (Stallings, 2003).

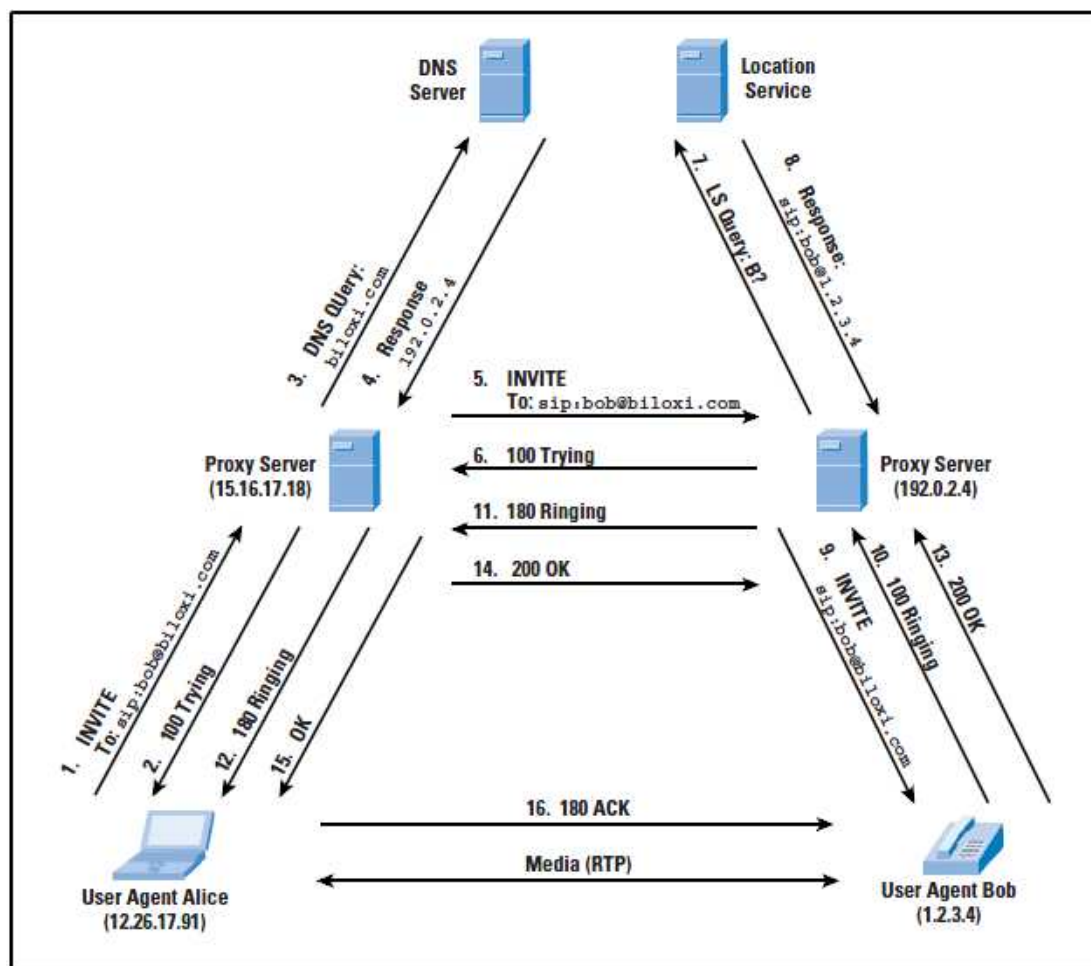


Figure 4.3 SIP session establishment (Stallings, 2003).

The DNS server responds (4) with the IP address of the **biloxi.com** proxy server (the inbound server). Alice's proxy server can now forward the INVITE message to the inbound proxy server (5), which acknowledges the message (6). The inbound

proxy server now consults a location server to determine Bob's location Bob (7), and the location server responds with Bob's location, indicating that Bob is signed in, and therefore available for SIP messages (8). The proxy server can now send the INVITE message on to Bob (9). A ringing response is sent from Bob back to Alice (10, 11, 12) while the UAS at Bob is alerting the local media application (for example, telephony). When the media application accepts the call, Bob's UAS sends back an OK response to Alice (13, 14, and 15) (Stallings, 2003).

Finally, Alice's UAC sends an acknowledgement message to Bob's UAS to confirm the reception of the final response (16). In this example, the ACK is sent directly from Alice to Bob, bypassing the two proxies. This occurs because the endpoints have learned each other's address from the INVITE/200 (OK) exchange, which was not known when the initial INVITE was sent. The media session has now begun, and Alice and Bob can exchange data over one or more RTP connections (Stallings, 2003).

4.4 SIP Presence Scenario

This example (Figure 4.4) makes use of two message types. These message types support telephony applications. Suppose that in the preceding example, Alice was informed that Bob was not available. Alice's UAC can then issue a SUBSCRIBE message (1), indicating that it wants to be informed when Bob is available (Stallings, 2003).

This request is forwarded through the two proxies in our example to a PINT (PSTN-Internet Networking) server (2, 3). A PINT server acts as a gateway between an IP network from which comes a request to place a telephone call and a telephone network that executes the call by connecting to the destination telephone. In this example, we assume that the PINT server logic is collocated with the location service. It could also be the case that Bob is attached to the Internet rather than a PSTN, in which case the equivalent of PINT logic is needed to handle SUBSCRIBE requests. In this example, we assume the latter and assume that the PINT

functionality is implemented in the location service. In any case, the location service authorizes subscription by returning an OK message (4), which is passed back to Alice (5, 6). The location service then immediately sends a NOTIFY message with Bob's current status of not signed in (7, 8, 9), which Alice's UAC acknowledges (10, 11, 12) (Stallings, 2003).

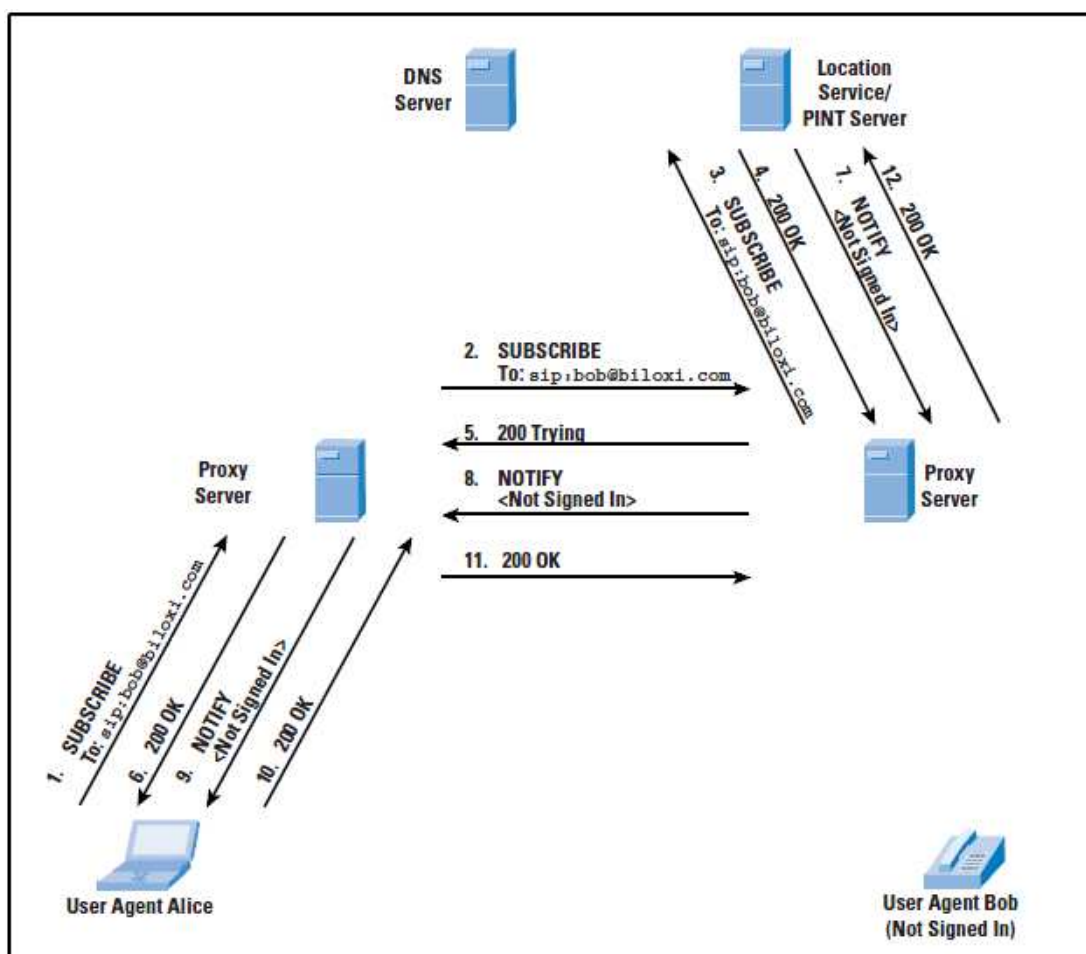


Figure 4.4 SIP presence example (Stallings, 2003).

Figure 4.5 continues the example of Figure 4.4. Bob signs on by sending a REGISTER message to the proxy in its domain (1). The proxy updates the database at the location service to reflect registration (2). The update is confirmed to the proxy (3), which confirms the registration to Bob (4). The PINT functionality learns of Bob's new status from the location server (here we assume that they are collocated) and sends a NOTIFY message containing Bob's new status (5), which is forwarded

to Alice (6, 7). Alice's UAC acknowledges receipt of the notification (8, 9 and 10) (Stallings, 2003).

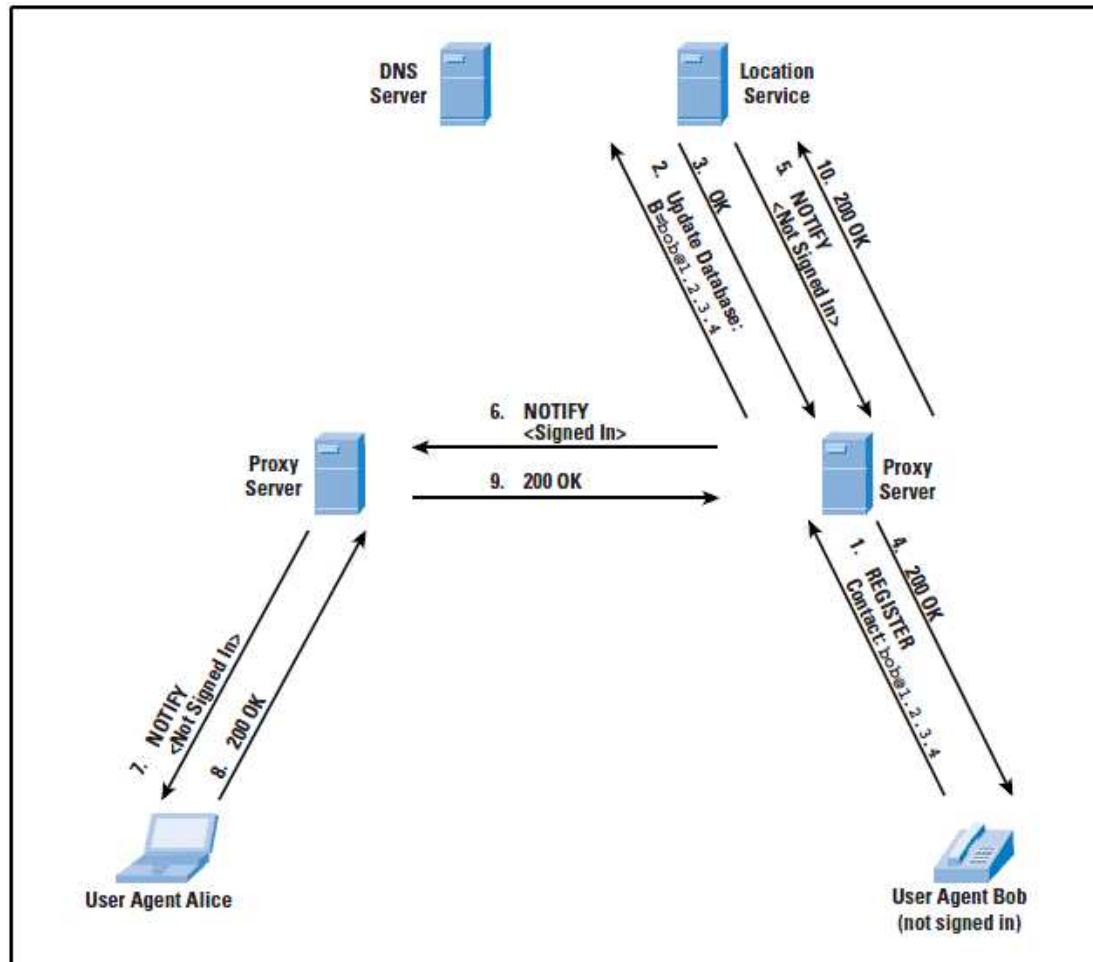


Figure 4.5 SIP registration and notification example (Stallings, 2003).

CHAPTER FIVE

VOIP PROTOTYPE

5.1 Introduction

A VoIP phone system requires the use of SIP phones / VoIP phones. SIP phones come in several versions/types:

- **Software based SIP phone:** it is a software based SIP phone is a program which makes use of your computer's microphone and speakers, or an attached headset to allow you to make or receive calls. Examples of SIP phones are eyeBeam from CounterPath (formerly Xten), OpenWengo, Nexge, sipXphone, Adore Softphone, Express Talk and SJphone. However, Sipdroid and Peers are the SIP softphone applications. Sipdroid works on the Android mobile phone or tablet and Peers works on Windows, Linux and Mac. These applications are examined and used as a SIP client in the VoIP system prototype which was implemented for this thesis.
- **USB VoIP phones:** A USB phone plugs into the USB port of a computer and with the use of a SIP/ VoIP soft phone software behaves just like a phone. Essentially it is not more than a microphone with a speaker, however because they appear like a normal phone they are more intuitive to use for a user.
- **Hardware SIP Phone:** A hardware based SIP phone looks like and behaves just like a normal 'phone'. However it is connected directly to the data network. These phones have an integrated mini hub, so that they can share the network connection with the computer. That way you do not need an additional network point for the phone. Examples of hardware SIP phones are Cisco, Linksys, Aastra, Snom and Grandstream.
- **Use analog phone via an ATA adapter:** If you want to use your current phone with the VoIP phone system, you can use an ATA adapter. An ATA adapter

allows you to plug in the Ethernet network jack into the adapter and then plug the phone into the adapter. That way your old phone will appear to the VoIP phone system software as a regular SIP phone.

5.2 Prototype Design

In order to implement VoIP prototype, an Android mobile phone with SipDroid application and a PC with Peers application are used as a user agent. Also, in the system two autonomous access points (AIR-AP1131AG) are used to allow these wireless clients to access a local area network. By using these access points, three types of wireless network configurations can be implemented. These wireless modes and the role of the access points in these modes are listed at below.

- Root access point: It is connected to a wired LAN and it supports wireless network to the clients.
- Repeater access point. It is not connected to a wired LAN and it associates to a root access point, and supports wireless clients.
- Workgroup Bridge: It is not connected to a wired LAN. It associates to a root access point or bridge and supports wired network devices.

In this thesis, the access points are used as root units. An autonomous access point is connected directly to a wired LAN and so, it provides a connection point for wireless user agents (Android mobile phone and PC). Since more than one autonomous access point is connected to the LAN, the user agents can roam one area of a facility to another without losing their connection to the network. As users move out range of one access point, they automatically connect to the associated network through another access point. The roaming process is seamless and transparent to the user. The implementation of the VoIP prototype is illustrated in the Figure 5.1.

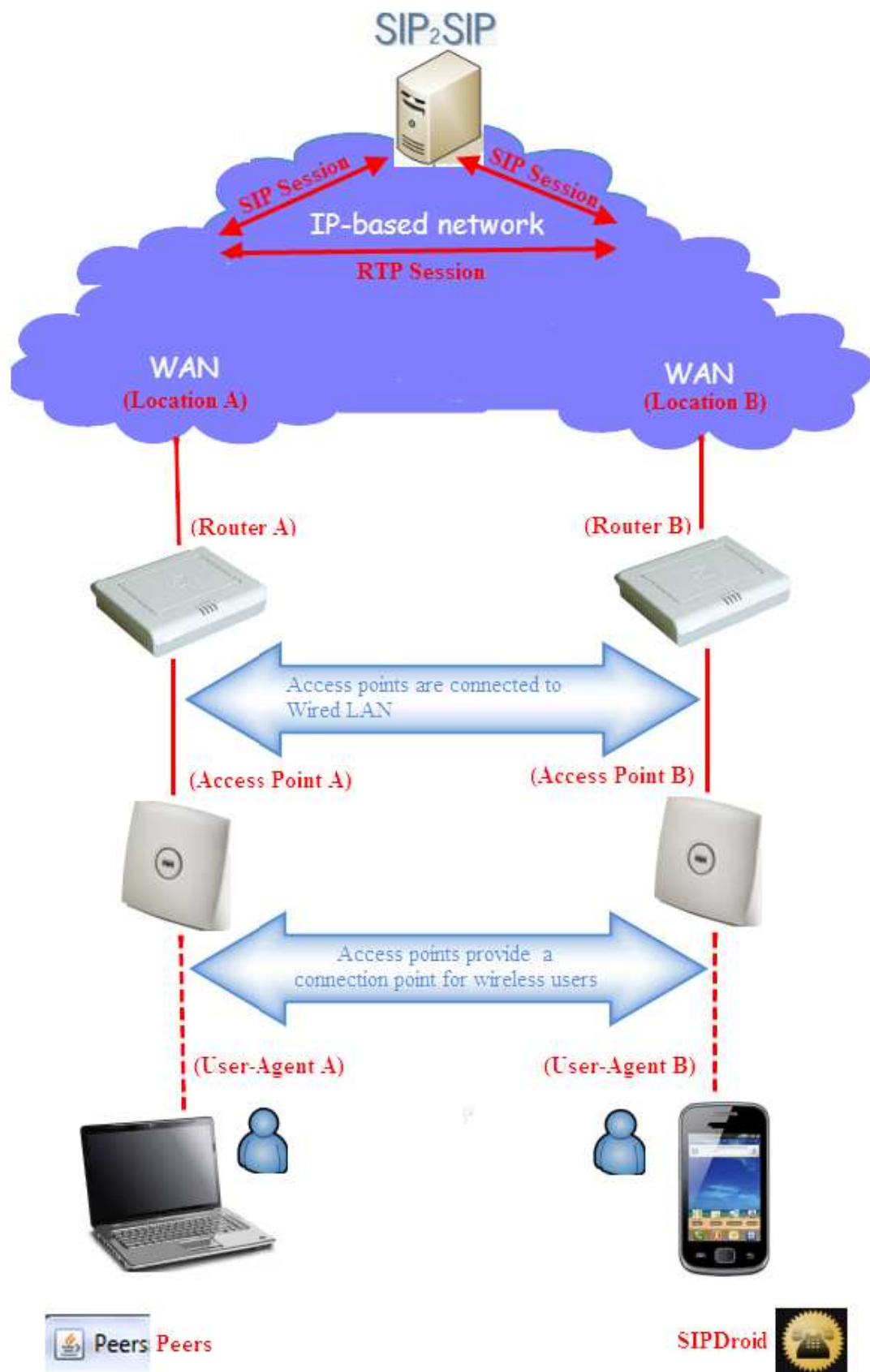


Figure 5.1 VoIP prototype.

5.3 Prototype Components

The main components which are used in the VoIP prototype include two SIP softphone applications which are called as Sipdroid and Peers and two access points which are used to provide wireless network for each SIP clients. The detailed information about these components is given in this chapter.

5.3.1 *Sipdroid*

Sipdroid is a java based, open source SIP client that has recently been developed for use with mobile devices based on Google's Android platform (Andrews, 2009).

Utilizing an SIP strategy using the Android is incredibly easy with Sipdroid, because Sipdroid will allow users to call any SIP-based PC. Initiating a session via PC will allow users to connect from their Android to a computer, which is definitely a good thing if a user wants to talk to someone who's sitting at their computer. But in reality, Sipdroid application is once set up into the mobile Android device, it will be able to connect with any SIP-enabled device, including other Androids.

These features allow for both voice and video conferencing, both of which are of course hallmarks of SIP communication and mobile communication in general. But these features also make it easy to have someone remotely connect to a meeting via SIP. Other traditional SIP capabilities can also be handled from the Android, allowing a new range of mobility for any business traveler or traveler in general.

Sipdroid allows users to use Android phone with almost any SIP provider. The calls are crystal clear even over a 3G network, likewise for a WLAN connection. The best part about Sipdroid is that it integrates into the phone, eliminating the need for a separate phone book. The users simply use their Android contacts in the same way as if they were making a call over a cellular network.

Although Sipdroid will likely mature quickly, it is currently only fully supported using virtual PBX service from PBXes.com. PBXes.com offers a free basic account registration for their service. Once a basic account with PBXes.com is created, additional SIP providers/registrars can be set up within the Trunks section of their web based UI. This service is free, but the set-up is a bit tricky.

Also, there are many free SIP providers such as SIP2SIP, GetOnSIP and Ekiga.net. Some of these SIP providers actually work without the use of <http://pbxes.org> and offer free voice service for only internet calls, not towards fixed line phone, nor mobiles.

In this thesis, Sipdroid application is executed by using a SIP account which is provided from one of the SIP providers called as SIP2SIP.

SIP2SIP is a free SIP service that provides a SIP account that can be used for applications beyond Voice over IP. The service is distributed in multiple data centers for high availability and scalability purposes. This service may be used to communicate using audio, video and instant messaging using the SIP protocol. The SIP account is Presence/IM enabled with support for relevant SIP SIMPLE standards for MSRP relay extension, Presence Agent, XCAP and RLS services. The features of SIP2SIP service are;

- Public SIP address under @sip2sip.info
- Audio and Video (RTP, sRTP, zRTP)
- IM and File Transfers (MSRP relay support)
- Presence (PUBLISH, SUBSCRIBE, NOTIFY)
- Contacts Management (XCAP protocol)
- NAT traversal including ICE support
- Conferencing for IM and Wideband audio

There are thousands of SIP devices on the market and SIP account on each device is configured differently. Each SIP provider will give user at least three parameters:

username, domain and password. Sometimes a SIP provider may give user an outbound proxy or a proxy.

The necessary information to create SIP account on Android Sipdroid application is written in the Table 5.1.

Table 5.1 Android Sipdroid configuration

Authorization Username	sip2sip username
Server or Proxy	proxy.sipthor.net
Domain	sip2sip.info
Password	xxx

In order to test this application and also to use as a SIP client in the VoIP system prototype, the open source code of the application is obtained. This code is examined in Eclipse Java Development Tool. In order to compile successfully, some optimizations and modifications are made in the source code. After compile operation, this SIP softphone application is loaded into the Android mobile phone. Before running the Sipdroid program, the necessary adjustments should be done on the application Settings screens.

The first step is creating SIP account which is provided from sip2sip service for the Sipdroid program. The SIP account information which is written in Table 5.1 should be entered into specified columns on ‘SIP Account’ screen. Since the aim of this thesis is to implement VoIP call over wireless networks, only WLAN technology is preferred to connect when the Sipdroid application initiates. The screenshots of these settings screens are shown in Figure 5.2 and Figure 5.3. After the completion of configuration, the application can be initiated.

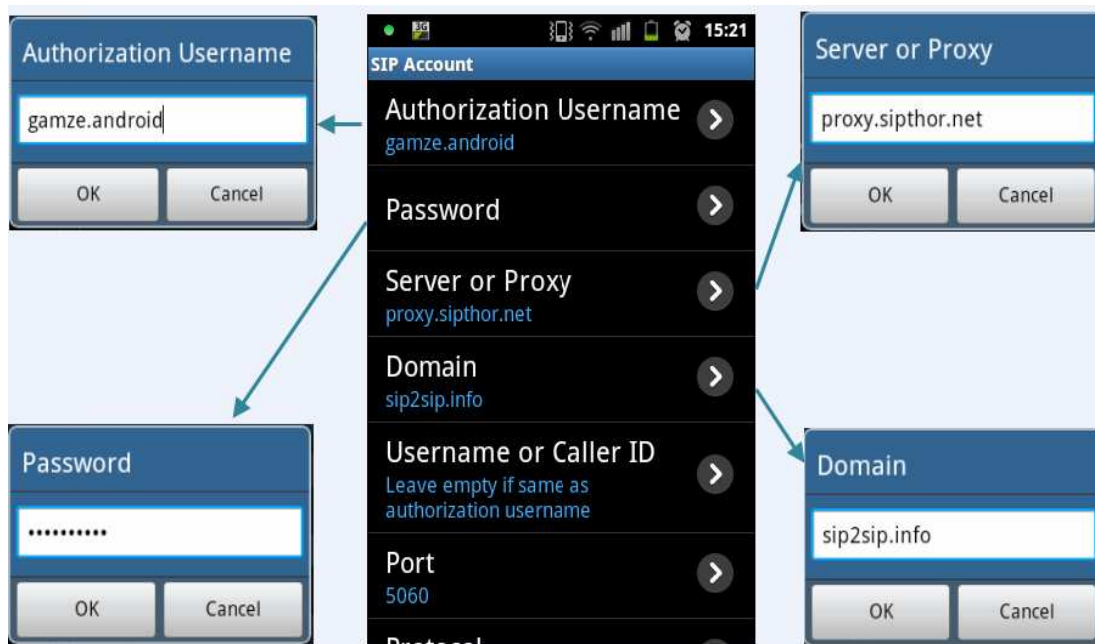


Figure 5.2 Settings screen of Sipdroid application.

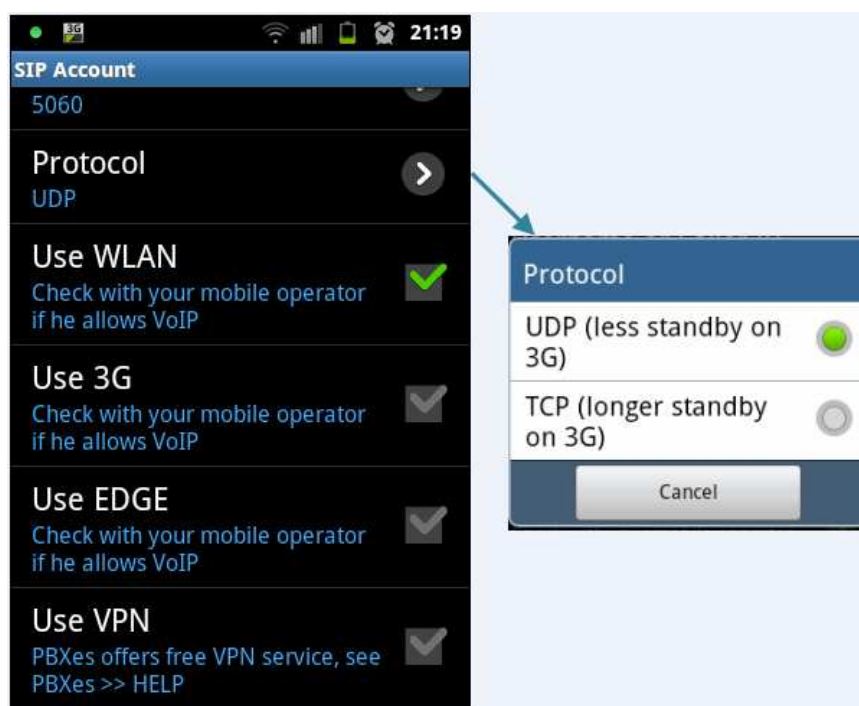


Figure 5.3 Settings screen of Sipdroid application.

5.3.2 Peers

Peers is an example of softphone (a software phone). To place calls using software, rules and unique “numbers” need to be established so that each software phone can place calls to other phones. Those rules are called a protocol. Although there are many protocols for voice communication over internet (VoIP) such as MGCP, H.323 and SIP, Peers softphone supports only SIP protocol.

Peers is written in java and works on Windows, Linux and Mac. It can be used with SIP servers like open sips or asterisk IPBX. It supports G711 codec (PCMU and PCMA) and telephone-events (DTMF).

In order to use this application, each user needs its own SIP account. SIP account for Peers application can be created from any free SIP provider. SIP account for this application is provided by the SIP provider (SIP2SIP) which is used for also Sipdroid application and mentioned in Section 5.3.1.

5.3.2.1 Architecture

Peers has been developed in java by using an object-oriented programming language. Peers is separated in packages (gui, sip, core, etc.). It relies only on standard java specification API and two java extensions APIs called as javasound and swing. If a standard java platform (opensdk, sunsdk) is used, everything is already included in this environment; there is no need to any library. The Peers architecture is shown in Figure 5.4.

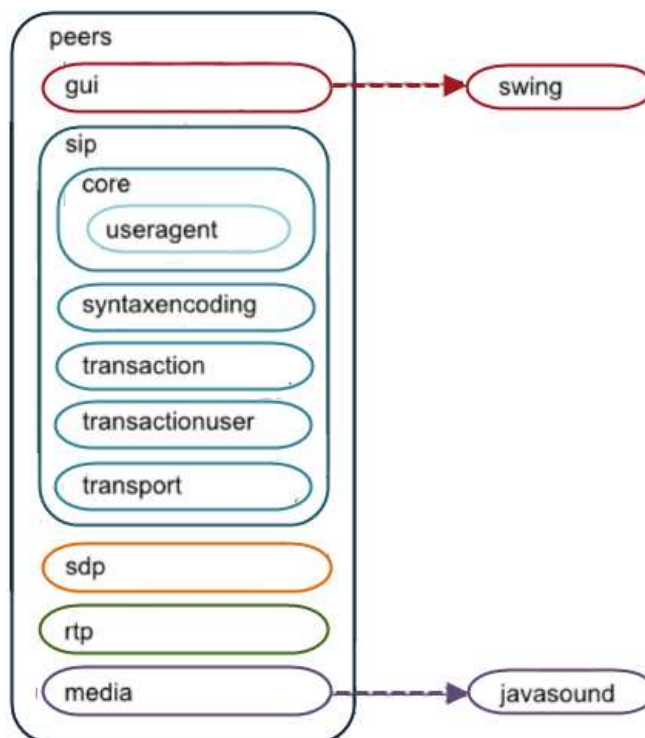


Figure 5.4 Peers architecture.

The first categorization in the source code is done on protocols and very high level capabilities:

- `net.sourceforge.peers.gui`
- `net.sourceforge.peers.media`: `media` package is responsible for sound encoding.
- `net.sourceforge.peers.sdp`: SDP does not rely on any external library. SDP package contains SDP related sources.
- `net.sourceforge.peers.sip`: SIP does not rely on any external library. SIP package contains SIP stack implementation. It is the only complicated package.

SIP stack is illustrated in Figure 5.5 and is made of:

1. `net.sourceforge.peers.sip.core`
2. `net.sourceforge.peers.sip.transactionuser`
3. `net.sourceforge.peers.sip.transaction`
4. `net.sourceforge.peers.sip.transport`
5. `net.sourceforge.peers.sip.syntaxencoding`

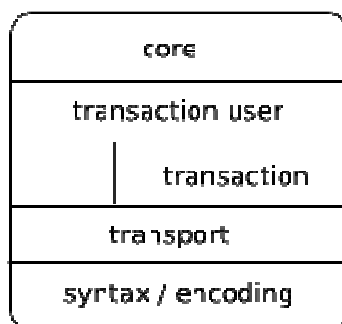


Figure 5.5 SIP stack

- `net.sourceforge.peers.rtp`: RTP package contains peers RTP implementation, which is used by media package.

5.3.2.2 SIP Package Details

Peers source code is separated in packages that correspond to SIP layers.

As mentioned in section 4.2, SIP uses two types of messages: requests and responses. Requests contain a method that will give request aim and a request for the person/server we want to reach. And responses contain a status code that gives response status: success, failure, etc. The detailed information about SIP messages are given in section 4.2. SIP messages have been separated in objects as shown in Figure 5.6 to ease message content access in Peers (Mantineau, 2011).

In Peers, transport package is quite simple. `TransportManager` creates client transports and server transports. Those client transports and server transports are called message senders and message receivers. Actually, behind the stage, `DatagramSockets` are doing the real job. The transport layer is also generally responsible for message retransmissions. As SIP works over UDP, those message retransmissions are very important to avoid losing messages (Mantineau, 2011).

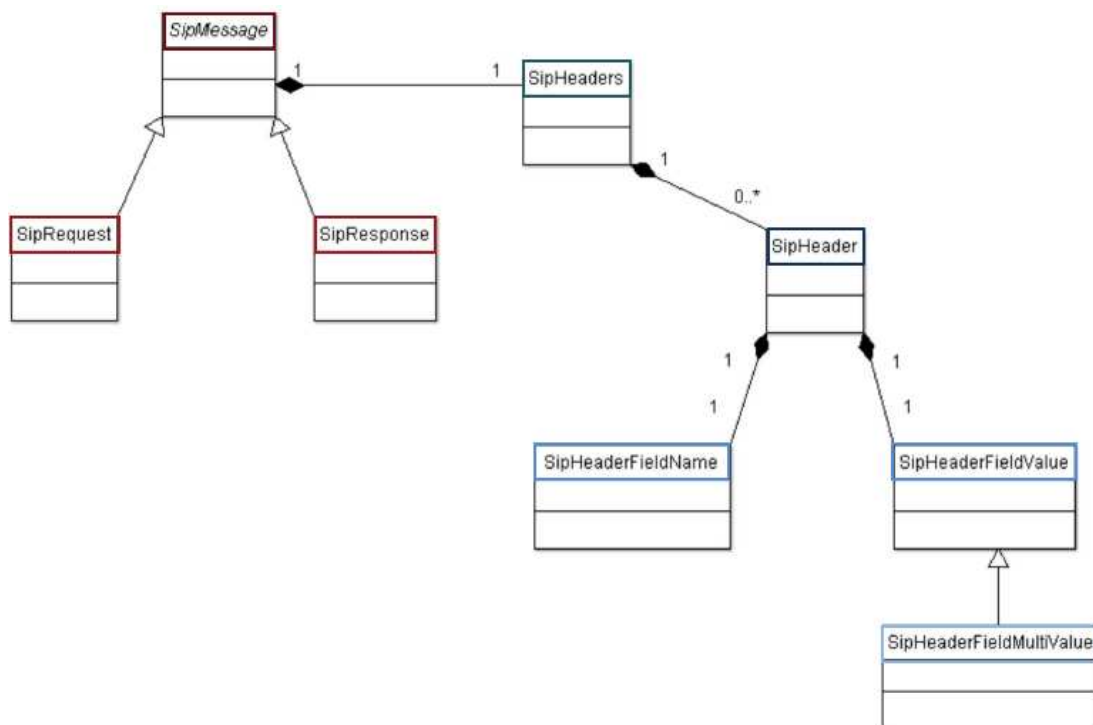


Figure 5.6 SIP message and its components.

Transport management has been done in a very naive way. In theory, UDP packets may contain several SIP messages, but this feature is not implemented in Peers. Actually on client side, this would probably be very odd to receive several SIP messages in the same UDP packet. In day-to-day life, it never occurs. Several multi-SIP messages UDP packets generally only occur between high-loaded servers, not on User-Agents. In theory, in SIP messages bigger than MTU or 1300 bytes if MTU is unknown are supposed to be sent on a reliable transport protocol such as TCP. In peers, this feature is not implemented and so, all packets are sent over UDP (Mantineau, 2011).

In summary, the requests are routed by using Route header and request-uri domain name. If Route header is not in message, IP address is used for routing requests. Responses are routed using via header. It generally contains an IP address and a port on which the response must be sent (Mantineau, 2011).

The other layer in SIP stack is transaction. The aim of the transaction can be described as “if everything goes well, else do nothing”. If any error occurs during

transaction management, before transaction management, modifications are aborted on transaction-related objects (generally dialog state, etc.) and the original state is come back. In SIP, a transaction is made of: exactly one request, eventually one or several provisional response(s) (status code between 101 and 199) and exactly one final response (status code between 200 and 699) (Mantineau, 2011).

Transactions that receive requests are called server transactions and transactions which send requests are called client transactions. Transactions that create a dialog are called invite transactions, as INVITE is the only method that can create dialogs in RFC3261. Transactions that will not create a dialog are called non-invite transactions. Thus there are four transaction types: invite client transaction, invite server transaction, non-invite client transaction and non-invite server transaction (Mantineau, 2011).

Server and client aspects of transaction have been implemented as interfaces in Peers, and invite and non-invite property have been implemented in abstract classes. Thus those four transactions have been implemented in their own class in Peers, extending and implementing the appropriate class and interface. Transaction manager works with transactions using their client/server property. Thus it uses ClientTransaction and ServerTransaction interfaces to handle them (Mantineau, 2011).

Several layers are using transaction layer on the upper side: core and dialog. Core is User-Agent, Proxy, Registrar or Redirect Server and dialog is transaction user. Transaction user is probably the simplest layer in SIP. It contains Dialogs. A dialog is the representation of a media session on the control side. There are two sides in SIP: media and control. Dialog is on control side and media session is on media side. Media session is often the term employed in SDP and RTP. One state machine is necessary for dialogs. Inside a dialog, there is information of local and remote contact addresses, unique id, etc. Dialogs are managed using DialogManager (Mantineau, 2011).

On the top of transaction user layer, core layer that defines the SIP element role is found. As already mentioned in Section 3.2.1.2, there were several nodes: proxy, registrar, redirect server and user-agent.

Peers is a user-agent. It's the software employed by users to place or receive calls. Actually, a user-agent is just the SIP part of this software. Since user-agent is considered as the image of the software in SIP stack, the corresponding package is named as 'net.sourceforge.peers.sip.core.useragent'. Peers SIP core layer or core role is User-Agent.

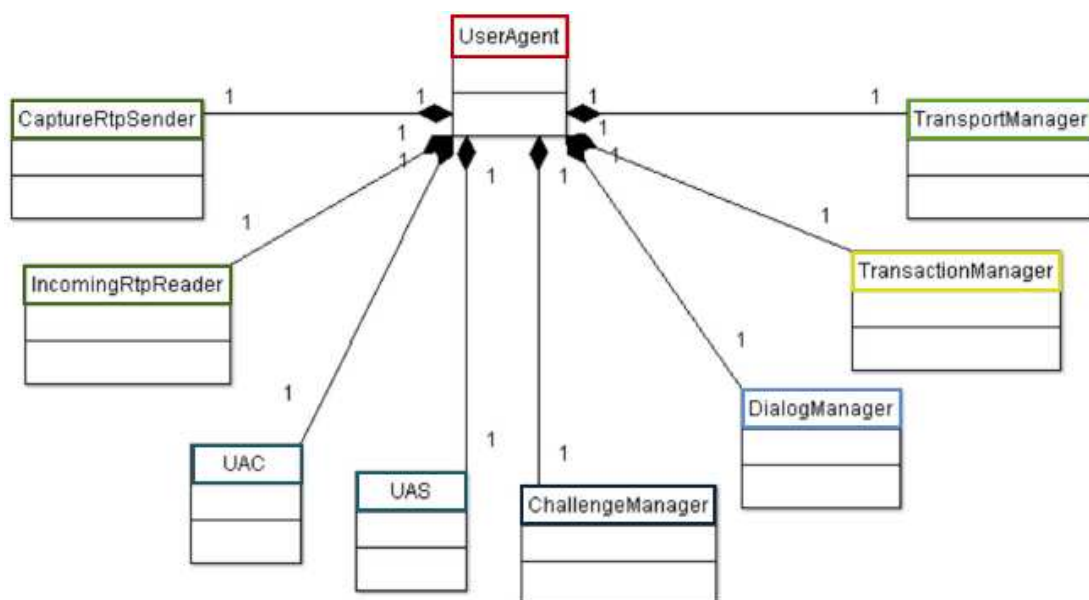


Figure 5.7 User-agent references.

In SIP, the core layer is the brain. Depending on its role, it can be more or less sophisticated, but it's the place where general behavior is defined. Another property of SIP protocol is that complex things are managed in client software applications. The complexity is implemented on the edge of the network in SIP protocol. To support this complexity, each single feature has been implemented in a separate class in Peers (Mantineau, 2011).

A user-agent always contains both a user-agent client and a user-agent server. A user-agent client is responsible for requests sending and a user-agent server is

responsible for incoming requests processing. In Peers, they are called UAC and UAS (Mantineau, 2011).

User Agent keeps references to many important objects: UAC, UAS, media related objects (CaptureRtpSender and IncomingRtpReader), and managers (ChallengeManager, DialogManager, TransactionManager and TransportManager).

Each layer manager is referenced here. All core Handlers and Managers are instantiated within UserAgent constructor. Thus, its underlying layer objects are created implicitly when the user instantiates a new UserAgent. (Mantineau, 2011).

Communication between core SIP layer (User-agent) and graphical user interface (GUI) is done using an interface called SipListener. Thus, separation between SIP layers and user interface is clearly identified. The communication is done with GUI package, but this graphical user interface could be replaced with another GUI, a web interface, or even a console interface thanks to this generic way of communicating between sip core layer and upper layer (Mantineau, 2011).

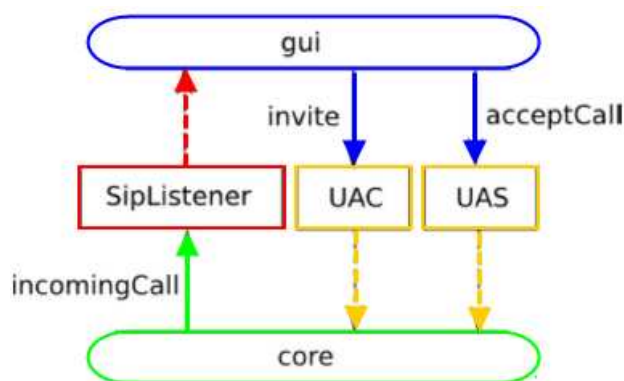


Figure 5.8 Interaction between core and GUI

The software which needs to use peers SIP stack with another GUI can instantiate a UserAgent object and then it can use this user agent instance to communicate with peers SIP stack. The SipListener interface offers the user interface a way to be notified of SIP incoming events: incoming call, callee pickup, remote hang up, etc. User actions are provided to sip core through UAC and UAS via

UserAgent.getUAC() and UserAgent.getUAS() methods: invite, register, acceptCall and rejectCall. Incoming events correspond to methods in SipListener interface. This interface is implemented by a main class in upper layer (EventManager in GUI package); the upper layer processes the incoming events and eventually takes action on SIP core layer depending on previous events and user actions (Mantineau, 2011).

5.3.2.3 SDP Package Details

SDP package is responsible for codec negotiation. The negotiation principle is quite simple. At any time, an entity generates an offer with all supported codecs. Firstly, this offer is sent to another entity. Then, the entity that received the offer parses this offer, analyzes it and finally generates an answer. There is always one answer for one offer. The answer is not always the same since it depends on offer (Mantineau, 2011).

In SIP theory, an offer can be present in either INVITE or 200 body. If the offer is in INVITE, the answer is in 200. If the offer is in 200, the answer is in ACK body and INVITE body is empty. Peers does not support empty INVITE body. SDP contains critical information about media streams. It provides the IP address and the port on which it wishes to receive RTP packets. It also describes the payload types that it supports. SDP gives media description, not media content. The protocol that transports media streams is RTP. This protocol transports encoded media with a specific wrapping format, defined as payload type (Mantineau, 2011).

In Peers source code, SDPManager is the place where everything is done at SDP level. This class generates offers, parses answers to extract useful information (IP address, port, payload type) and generates answers based on incoming offers. The object employed to describe an SDP body is SessionDescription. A SessionDescription can contain several MediaDescriptions. A MediaDescription typically corresponds to an audio stream or a video stream. A MediaDescription can contain several Codecs. SDP objects are illustrated in Figure 5.9. MediaDestination

is the combination of IP address, port and Codec. This class eases RTP targets description (Mantineau, 2011).

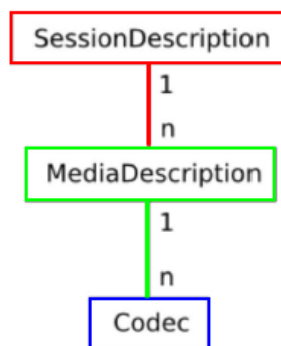


Figure 5.9 SDP objects.

5.3.2.4 Media Package Details

The main classes of this package are responsible for RTP packetization/depacketization, media encoding, media decompression, and microphone capture and media playback. The media package relies on RTP peers package and javasound (Mantineau, 2011).

Javasound is standard sun javasound API. The use of javasound for media capture and playback is critical. Although it is not the simplest java media API, it has the advantage of being tested by sun on each supported platform (Windows, Linux, Mac, Solaris, etc.). Peers has been tested successfully on Linux, Windows and Mac OS 10.6. Even though Javasound has many drawbacks such as few guaranteed features and no standard audio data format, it is already integrated in java standard edition API. Also, it avoids third-party libraries with native parts (Mantineau, 2011).

Peers codecs have been implemented the most generic way as possible. As PCMU and PCMA are supported in peers, each one has its own Encoder and Decoder. Encoder and Decoder classes in Peers application only define a process method that will work on input to generate an output depending on codec algorithm. Encoder is started at the beginning of a call and only the codec-specific encoding is done in concrete class (Mantineau, 2011).

5.3.2.5 RTP Package Details

RTP stack in Peers is very simple. The main class of this stack is `RtpSession`. This class handles the packet sending and reception. It relies on java 5 `ExecutorService` to receive packets. Nevertheless it does not create a separate thread to send packets; it simply sends them on demand using `send (RtpPacket)` method. As other packets, RTP package makes use of a listener to notify the reception of an RTP packet. When a packet is received, the raw data is parsed using `RtpParser` class. Thus a new `RtpPacket` is created and provided to the `RtpListeners` which subscribed to RTP packet reception on `RtpSession` (Mantineau, 2011). RTP packet flow is illustrated in Figure 5.10.

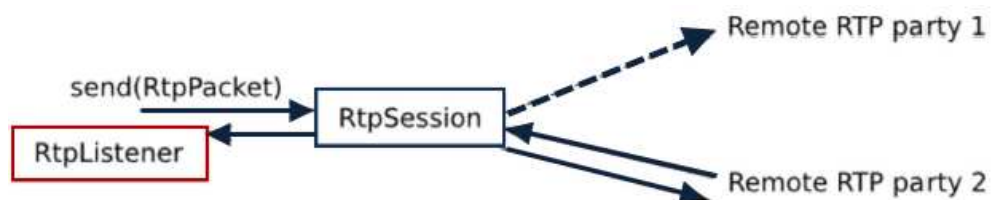


Figure 5.10 RTP packet flow.

An `RtpPacket` contains standard RTP headers and data (SSRC, SequenceNumber, PayloadType, etc.). Peers RTP stack is based on RFC3550. Before it can be used, an `RtpSession` has to be started using ‘`start()`’ method. Later it can be stopped using ‘`stop()`’ method. An `RtpSession` uses an initial remote IP address and a port number to send first RTP packets before any RTP packet has been received from remote RTP party. Once a packet has been received, if the IP address or the port from which the packet is coming differs from the previous ones, the remote IP address and port are updated with latest ones. This enables NAT traversal in a few cases. Thus peers RTP stack support symmetric RTP (Mantineau, 2011).

5.3.2.6 GUI Package Details

Peers is based on swing for GUI management. GUI is based on a class which manages all events coming from SIP layer and from user and dispatches events among GUI components. This class is `EventManager`. This class receives invocations

from SIP layer using its SipListener interface. Then it dispatches events to the appropriated frames. EventManager is listening to frames through listener's interfaces: MainFrameListener, CallFrameListener. EventManager also implements ActionListener to receive events from MainFrame JMenu. Thus when user is clicking on Edit > Account, the corresponding event reaches EventManager and EventManager eventually instantiates a new AccountFrame to enable user configure SIP account. Several frames are implemented in GUI (Mantineau, 2011).

The first one is MainFrame. It is the most important and first class to be instantiated. Mainframe is actually the class that contains the main() method of peers. It creates and references EventManager. Then EventManager creates and manages other frames and their corresponding events. The screenshot of MainFrame is shown at Figure 5.11.



Figure 5.11 Main frame in action.

The next one is CallFrame. When user places or receives a new call, a new CallFrame is created and displayed to user. The screenshots of CallFrame before and after callee pickup are shown in Figure 5.12 and Figure 5.13.



Figure 5.12 Call frame (before pickup) in action.



Figure 5.13 Call frame (after pickup) in action.

The last one is AccountFrame. This frame enables peers user to configure his or her SIP account with a minimal interface as shown in Figure 5.14.



Figure 5.14 Account frame in action.

This class also displays the registration state of the corresponding account. Actually, this registration state is also displayed in MainFrame to let user know that his or her account is registered on regular Peers startup.

5.3.3 Wireless Access Points

The function of a wireless access point is to allow wireless devices such as projectors, PCs and PDAs to access a local area network. Wireless access points mainly act as switches to spread connections wirelessly. The difference between an

access point and a router is that access points do not assign IP addresses and they don't have firewalls. Access points only lock out traffic that does not have the wireless key.

Wireless access points (APs or WAPs) are specially configured nodes on wireless local area networks (WLANs). Access points act as a central transmitter and receiver of WLAN radio signals. Access points used in home or small business networks are generally small, dedicated hardware devices featuring a built-in network adapter, antenna, and radio transmitter. Access points support Wi-Fi wireless communication standards. Although very small WLANs can function without access points in so-called "ad hoc" or peer-to-peer mode, access points support "infrastructure" mode. This mode bridges WLANs with a wired Ethernet LAN and also scales the network to support more clients. Older and base model access points allowed a maximum of only 10 or 20 clients; many newer access points support up to 255 clients (Mitchell, n.d.).

5.3.3.1 Cisco Aironet 1130AG Series Access Point

Cisco Aironet 1130AG Series Access Point (models: AIR-AP1131AG and AIR-AP1131G) supports a management system based on Cisco IOS software. The 1130AG series access point is a Wi-Fi certified, wireless LAN transceiver. The 1131AG access point uses dual integrated radios (IEEE 802.11g and IEEE-802.11a). The access point serves as the connection point between wireless and wired networks or as the center point of a stand-alone wireless network. In large installations, wireless users within radio range of an access point can roam throughout a facility while maintaining uninterrupted access to the network (Cisco Systems, 2008).

In this thesis, AIR-AP1131AG model access points are used in order to implement the VoIP system prototype. Installation procedure is applied to access points by following "Quick Start Guide Cisco Aironet 1130AG Access Point" document. First of all, the access point is connected to power source. When power is supplied to the access point, a routine power-up sequence began by changing the

access point status LED. During the power up sequence the LED displayed a series of colors. When the power up sequence was complete, the LED displayed a light green color. This color indicates that it is ready for operation. At this point, the access point needs an IP address to operate. The access point is no longer shipped with a default IP address. The existing network had a DHCP server and so, it obtained an IP address from existing network's DHCP server when the access point was connected to the corresponding network. If the existing network does not have a DHCP server, the access point continues to request an IP address until it is assigned one. In this case, the IP address must be configured by opening the command line interface (CLI) from a terminal session established through the access point's console port. The scope of this thesis, IP address is obtained from DHCP server automatically. Since the access point obtained its IP address from the network's DHCP server, assigned IP address is found by querying the DHCP server using the access point's MAC address.

After the IP address is obtained, in order to configure basic settings, the PC which is in the same network with the access point is used. IP address is entered into web browser address field. After the correct username/password ('Cisco/Cisco' as a default) is entered into pop up window, the settings screen is opened.

The necessary configurations are implemented in the corresponding settings windows. The screenshots of configuration screens are shown in Figure 5.15 and 5.16.

Hostname **ap1** ap1 uptime is 31 minutes

Express Set-Up

Host Name: It is a name for the access point that identifies it on the existing network

MAC Address:

Configuration Server Protocol: ☒ DHCP ☐ Static IP It specifies how the access point obtains an IP address. DHCP: IP address is automatically assigned by the network. Static IP: It uses an IP address that is entered in the IP address field.

IP Address: It is obtained from DHCP server or is assigned with static IP address.

IP Subnet Mask:

Default Gateway:

SNMP Community:

☒ Read-Only ☐ Read-Write

Radio0-802.11G

Role in Radio Network: ☒ Access Point ☐ Repeater ☐ Workgroup Bridge ☐ Universal Workgroup Bridge Client MAC:
☐ Scanner

Optimize Radio Network for: ☐ Throughput ☐ Range ☒ Default ☐ [Custom](#)

Aironet Extensions: ☒ Enable ☐ Disable

Radio1-802.11A

Role in Radio Network: ☒ Access Point ☐ Repeater ☐ Workgroup Bridge ☐ Universal Workgroup Bridge Client MAC:
☐ Scanner

Optimize Radio Network for: ☐ Throughput ☐ Range ☒ Default ☐ [Custom](#)

Aironet Extensions: ☒ Enable ☐ Disable

Figure 5.15 Express set-up screen.

Hostname **ap1** ap1 uptime is 1 hour, 4 minutes

Express Security Set-Up

SSID Configuration

It is a unique identifier that clients use to associate with the access point. It helps client devices distinguish between multiple wireless networks in the same vicinity

1. SSID ☐ Broadcast SSID in Beacon

2. VLAN

☒ No VLAN ☐ Enable VLAN ID: (1-4094) ☐ Native VLAN

3. Security

☐ No Security

☒ Static WEP Key

It is more secure than no security. There are two different lengths for WEP keys: 40 bit and 128 bit. Cisco access points use hexadecimal characters.

Key 1 40 bit

☐ EAP Authentication

RADIUS Server: (Hostname or IP Address)

RADIUS Server Secret:

☐ WPA

RADIUS Server: (Hostname or IP Address)

RADIUS Server Secret:

After the configuration is completed, 'Apply' button is clicked and then SSID record is added into table.

SSID Table

Delete	SSID	VLAN	Encryption	Authentication	Key Management	Native VLAN	Broadcast SSID
<input checked="" type="radio"/>	AP1	none	wep mandatory	open	none		✓

Figure 5.16 Express security set-up screen.

After the configuration procedure is completed, the access point is appeared in wireless network list as shown in Figure 5.17.

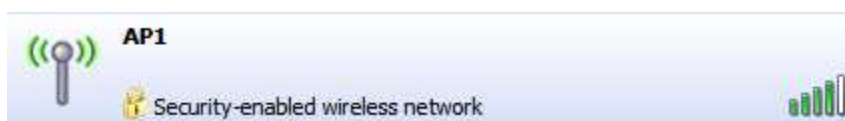


Figure 5.17 Access point in wireless network list.

CHAPTER SIX

CONCLUSION

6.1 Conclusion

The main idea of this thesis is to understand VoIP technology by examining general structure, working principles and fundamental components as well as to implement the simple VoIP prototype in wireless networks. The general structure of this technology is explained by comparing with the traditional circuit switched networks and by emphasizing its superior features. VoIP technology uses the Internet's packet-switching capabilities to provide phone service. Packet switching allows several telephone calls to occupy the amount of space occupied by only one in a circuit-switched network. Also, in packet switching, instead of routing the data over a dedicated line, the data packets flow through a chaotic network along thousands of possible paths.

In this thesis, the VoIP signaling protocols, H.323 (ITU-T standard) and SIP (IETF standard), are discussed and compared. Although both protocols are designed for VoIP applications, their original focus is very different. The focus of H.323 has been set to handle voice and multimedia calls including supplementary services. Otherwise, SIP has been designed as a generic transaction protocol for session initiation, not limited to any specific media services like audio or video. H.323 has more share of the market at present, but SIP is a much better protocol given its simplicity and scalability. Since in the public VoIP services, SIP is currently the dominant technology, SIP soft phone applications were preferred to use in the VoIP prototype. In order to do the actual voice transport, the VoIP system needs some real time protocols. An overview of real time protocols by specifying their functions are also stated. These protocols can be used with both H.323 and SIP protocols. RTP and RTCP are used for the real-time transport and controlling. RTSP is used to provide controlled delivery of media streams. RSVP is used to reserve resources in the network and thereby provide some Quality of Service. Also, some protocols are

required in conjunction with SIP so as to advertise the session (SAP) and give a description of the session (SDP).

The components of VoIP include end-user equipment, network components, gateways and servers. End-user equipment is used to access the VoIP system to communicate with another end point. Signaling servers handle the application level control of the routing of signaling messages to initiate a VoIP call. Connection to the network may be physically cabled or may be wireless. Since today's communication industry based on the mobility and easy access to the source, connection to the network was implemented via wireless technology. The end-user equipment may be a hard-phone or a softphone that is installed on a PC or a mobile phone. Softphones become more widespread in a little while since they are cheaper than hard-phones, also most of them are free applications which are easily loaded into smart phones and they have flexible and handy features accordance with modern time. Because of these reasons, SIP softphone applications are preferred in the VoIP prototype.

In the VoIP prototype, two different softphone applications, Sipdroid and Peers, are installed into an Android based mobile phone and a PC. In the both applications, SIP signaling protocol and RTP real time protocol are used. The network components such as cabling, routers, switches and access points are used in order to design VoIP prototype. The access points are connected to the wired network via switches. The aim of the access point usage is to obtain wireless network for soft phones. End users are connected to different wireless connection points which are provided by access points. For each soft phone application, a SIP account is created from the Sip2Sip SIP server. When the call procedure is run in these applications, SIP session establishment is setup via SIP signaling server. This signaling server handled the application level control of the routing of signaling messages in order to initiate a SIP call. After the completion of the session establishment, the data is transported via RTP real time protocol.

As a result, the prototype of VoIP system was implemented over wireless networks. The components and protocols which are used in the prototype are selected by considering their superior functions and compatibility with current technologies.

6.2 Future Works

In this prototype, the overall structure of VoIP system was examined in detail. VoIP call scenarios which were mentioned in the third chapter were observed in real time by using Wireshark network protocol analyzer. Although VoIP technology was analyzed in detail, the design principles and quality of service factors weren't discussed in the scope of this thesis. As a future work, this VoIP prototype should be examined by taking into consideration to quality factors such as delay, jitter and packet loss and the prototype should be improved by considering these issues. Also, Comparison between the two SIP softphone applications can be made by considering their performances over Wireless networks. Thus, advantages and shortcomings against each other can be specified.

REFERENCES

- Ahson, S. & Ilyas, M. (Eds.). (2009). *VoIP handbook applications, technologies, reliability, and security*. Boca Raton: CRC Press.
- Andrews, C. (April, 2009). *First look: SIPDroid open source SIP client for Android mobile phones*. Retrieved August 20, 2011, from <http://www.voipsupply.com/blog/first-look-sipdroid-open-source-sip-client-for-android-mobile-phones>
- Arora, R. (November 23, 1999). *Voice over IP: Protocols and standards*. Retrieved December 5, 2011, from http://www.cs.wustl.edu/~jain/cis788-99/ftp/voip_protocols.pdf
- Bakshi, M. (April 24, 2006). *VoIP / Multimedia over WiMAX (802.16)*. Retrieved December 25, 2009, from http://www.cse.wustl.edu/~jain/cse574-06/ftp/wimax_voip.pdf
- Berson, S. (March 5, 1999). *RSVP protocol overview*. Retrieved December 5, 2011, from <http://www.isi.edu/rsvp/overview.html>
- Beuran, R. (April 20, 2006). *VoIP over Wireless LAN survey*. Retrieved December 9, 2011, from http://www.starbed.org/~razvan/publications/voip_survey_final.pdf
- Cisco Aironet 1130AG Series access point hardware installation guide*. (June, 2008). Retrieved December 17, 2011, from http://www.cisco.com/en/US/docs/wireless/access_point/1130/installation/guide/1130hig_book.pdf
- Freeman, R. L. (Ed.). (2005). *Fundamentals of telecommunications* (2nd ed.). New Jersey: John Wiley & Sons, Inc..
- Hetawal, A. (2005). *Real Time Protocols, RTP, RTCP, RTSP*. Retrieved December 5, 2011, from www.cis.udel.edu/~amer/856/rtp.05f.ppt

- Igbal, N. & Cheema, F. M. (January, 2009). *QoS of VoIP in wireless networks*. Retrieved January 9, 2011, from <http://ebookbrowse.com/voip-in-wireless-environment-thesis-2009-doc-d170306765>
- Ismail, M. N. (March, 2011). *Analysis of VoIP softphone performance between wired and wireless in campus network environment*. Retrieved June 12, 2011, from <http://www.yangsky.com/ijcc/pdf/ijcc91/IJCC926.pdf>
- Johnston, A., Donovan, S. & Sparks, R. (December, 2003). *Session initiation protocol (SIP) basic call flow examples*. Retrieved September 12, 2011, from <http://www.ietf.org/rfc/rfc3665.txt?number=3665>
- Kashihara, S. (Ed.). (2011). *VoIP technologies*. India: InTech.
- Latif, T. & Malkajiri, K. K. (2007). *Adoption of VoIP*. Retrieved June 15, 2011, from <http://epubl.luth.se/1653-0187/2007/003/LTU-PB-EX-07003-SE.pdf>
- Liu, C. (January 15, 1998). *Multimedia Over IP: RSVP, RTP, RTCP, RTSP*. Retrieved September 12, 2011, from http://www.cs.wustl.edu/~jain/cis788-97/ftp/ip_multimedia.pdf
- Martineau, Y. (March, 2011). *Peers-documentation*. Retrieved May 8, 2011, from <http://peers.sourceforge.net/documentation>
- Mehdi, G. (November, 2009). *Future of VoIP over wireless in economic downturn*. Retrieved November 24, 2009, from [http://www.bth.se/fou/cuppsats.nsf/all/c3bf468f91c1eba8c1257672008055c6/\\$file/Ghazzal%20Thesis.pdf](http://www.bth.se/fou/cuppsats.nsf/all/c3bf468f91c1eba8c1257672008055c6/$file/Ghazzal%20Thesis.pdf)
- Minoli, D. (2006) *Voice over IPv6: architectures for next generation VoIP networks*. Oxford: Elsevier Inc.
- Mitchell, B. (n.d.). *Access point, wireless*. Retrieved December 15, 2011, from

http://compnetworking.about.com/cs/wireless/g/bldef_ap.htm

Schulzrinne, H. & Casner, S. (July, 2003). *RTP: A Transport Protocol for Real-Time Applications*. Retrieved December 5, 2011, from
<http://www.javvin.com/protocol/rfc3550.pdf>

Stallings, W. (2003). The session initiation protocol. *The Journal of Cisco the Internet Protocol*, 6 (1), 20-30. Retrieved December 5, 2011, from
http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-1/ipj_61.pdf

Sulkin, A. (2002). *PBX systems for IP telephony*. United States of America: McGraw-Hill Companies, Inc..

Tong, H. A. (2005). *SIP-based VoIP service – architecture and comparison*. Retrieved November 24, 2011, from
http://www.linecity.de/INFOTECH_ACS_SS05/acs5_top2_paper.pdf

Udani, S. & Mehta C. (February, 2001). *Overview of voice over IP*. Retrieved June 10, 2011, from
http://www.jdsu.com/voipterm-wp-acc-tm-ae/upen_Overview_VoIP.pdf