NAT TRAVERSAL AND MOBILITY IN VOIP APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE ATILIM UNIVERSITY

BY

NADIR AHMED GAYLANI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

September 2005

Approval of the Graduate School of NATURAL AND APPLIED SCIENCES.

Prof. Dr. İbrahim Akman
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of MASTER OF SCIENCE.

Prof. Dr. İbrahim Akman
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of MASTER OF SCIENCE.

Asst. Prof. Dr. Murat Erten
Co-Supervisor

Prof. Dr. İbrahim Akman
Supervisor

Examining Committee Members

Prof. Dr. İbrahim Akman _____

Asst. Prof. Dr. Murat Erten _____

Asst. Prof. Dr. Çiğdem Turhan _____

Instructor Bülent Gürsel Emiroğlu _____

Instructor Kasım Öztoprak _____

ABSTRACT

NAT TRAVERSAL AND MOBILITY IN VOIP APPLICATIONS

Gaylani, Nadir Ahmed

MSc, Department of Computer Engineering

Supervisor: Prof. Dr. İbrahim Akman

Co-Supervisor: Asst. Prof. Dr. Murat Erten

September 2005, 68 pages

Internet was initially designed to transfer data, but later it became common to use it for multimedia applications like VoIP, and with the increase of users, available Internet global addresses became limited which led to the use of NAT (Network Address Translation). In addition to that users started to move around while connected to the Internet

This affects the P2P applications like VoIP. In this work we have studied the effect of NAT and mobility on VoIP, and we proposed a solution that can be used in the presence of NATs with the mobile nodes using SIP signaling. Our solution shows that location servers are not necessary when the mobile node moves to a new location

Keywords: VoIP,NAT,SIP,Mobility

ÖZ

VOIP UYGULAMALARINDA NAT GEÇİŞİ VE DÜĞÜM HAREKETLİLİĞİ

Gaylani, Nadir Ahmed

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. İbrahim Akman

Ortak Tez Yöneticisi: Asst. Prof. Dr. Murat Erten

Eylül 2005, 68 sayfa

Internet ilk olarak data transferi için tasarlanmıs fakat sonra çokluortam uygulamaları için de yaygın olarak kullanılmaya başlanmıştır. Kullanıcıların çoğalması ile global Internet adresleri yetersiz kalmıştır. Bu sınırlamayı aşmak için NAT'lar kullanılmaya başlanmış ve kullanıcılar hareket halinde iken de Internet'e erişmeye başlamışlardır.

Bu sonuçlar VoIP gibi P2P uygulamalarını etkilemiştir. Bu çalışmada NAT kullanılmasının ve mobilitenin VoIP üzerindeki etkileri incelenmiş ve SIP sinyalleşmesine dayanan bir çözüm önerilmiştir. Cözümümüzde düğümün hareket etmesi halinde lokasyon sunucusuna erişim gerekmemektedir

Anahtar Kelimeler: VoIP,NAT,SIP,Mobility

To my Wife and Family

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisors Prof. Dr. İbrahim Akman and Asst. Prof. Dr. Y. Murat Erten for their guidance and insight throughout this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

3D Three Dimensional.

AbS Analysis.by.Synthesis

ACELP Algebraic Codebook Excited Linear Prediction

ADM Adaptive Delta Modulation

ADPCM Adaptive Differential Pulse Code Modulation

APCM Adaptive Pulse Code Modulation

API Application Programming Interface

ATM Asynchronous Transfer Mode

BGP Border Gateway Protocol

CELP Codebook Excited Linear Prediction

CNAME Canonical Name

CS.ACELP Conjugate Structure Algebraic Codebook Excited Linear Prediction

CSRC Contributing Source

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

DM Delta Modulation

DoD Department of Defence

DPCM Differential Pulse Code Modulation

FTP File Transfer Protocol

HRIR Head.Related Impulse Response

HRTF Head.Related Transfer Function

HTTP Hypertext Transfer Protocol

IGMP Internet Group Management Protocol

IETF Internet Engineering Task Force

IID Interaural Intensity Difference

IP Internet Protocol

IPv4 Internet Protocol version four

IPv6 Internet Protocol version six

IRC Internet Relay Chat

ISDN Integrated Services Digital Network

ISO International Standards Organisation

ITD Interaural Time Difference

ITU International Telecommunication Union

JMF Java Media Framework

JRTPLIB Jori's RTP Library

LAN Local Area Network

LD.CELP Low Delay Codebook Excited Linear Prediction

LPC Linear Predictive Coding

MCU Multipoint Control Unit

MOS Mean Opinion Score

MP.MLQ Multipulse Maximum Likelihood Quantisation

MPE Multipulse Excited coding

MTU Maximum Transfer Unit

N-ISDN Narrowband Integrated Services Digital Network

NTP Network Time Protocol

OSI Open Systems Interconnection

OSPF Open Shortest Path First

PCM Pulse Code Modulation

RAS Registration, Admission and Status

RELP Residual Excited Linear Prediction

RPE Regular Pulse Excited coding

RPE-LTP Regular Pulse Excitation-Long Term Prediction

RSVP Resource Reservation Protocol

RTP Real-time Transport Protocol

RTCP RTP Control Protocol

RTSP Real-Time Streaming Protocol

QoS Quality of Service

SCMP ST Control Message Protocol

SDES Source Description

SDP Session Description Protocol

SIP Session Initiation Protocol

SMTP Simple Mail Transfer Protocol

SSRC Synchronization Source

ST2 Stream Protocol Version 2

TCP Transmission Control Protocol

TTL Time To Live

UDP User Datagram Protocol

URL Uniform Resource Locator

VoATM Voice over ATM

VoFR Voice over Frame Relay

VoIP Voice over IP

VSELP Vector Sum Excited Linear Prediction

WAN Wide Area Network

CHAPTER 1

Introduction

In the last years with the spread of the Internet and with the large bandwidth that became available the opportunity of using this bandwidth for other purposes other that data became possible. Voice over IP is one of these.

VoIP is the technology for transporting integrated digital voice, video and data over IP networks. It is considered by some as the next evolutionary step in the digital transmission of data.

1.1 VoIP Protocols

There are two protocols that become dominant for voice over IP. They are:

- H.323 which is a globally accepted ITU standard for audio/video/data communication. It specifically describes how multimedia communications occur between user terminals, network equipment, and assorted services on Local and Wide Area Internet Protocol (IP) networks.

- SIP which is part of the IETF standards process and has been modeled upon other Internet protocols such as SMTP (Simple Mail Transfer Protocol) and

HTTP (Hypertext Transfer Protocol.) It is used to establish, change/modify and tear down (end) calls between one or more users in an IP-based network.

H.323 is the more mature of the two, but problems may arise due to lack of flexibility. SIP is currently less defined, but has greater scalability which could ease Internet application integration which give it the chance of becoming the more preferred protocol

1.2 NAT and VoIP

As the amount of information and resources increases, it is becoming a requirement for even the smallest businesses and homes to connect to the Internet. Network Address Translation (NAT) is a method of connecting multiple computers to the Internet (or any other IP network) using one IP address. This allows home users and small businesses to connect their network to the Internet cheaply and efficiently [36].

The impetus towards increasing use of NAT comes from a number of factors:

- A world shortage of IP addresses.
- Security needs.
- Ease and flexibility of network administration.

VoIP is one of many applications that can be adversely affected when IP clients connect through a NAT or NAPT. The NAT device may use an application level gateway (ALG). An ALG examines and modifies application payload content to allow packets from a specific application or protocol to pass through the NAT transparently. However, few NAT devices offer ALG functions for VoIP, and some protocols are not amenable to this approach. There are several categories of problems that VoIP applications have with NAT [14].

1. Many applications fail with NAT because the packets contain IP address or port information in the payload. A simple NAT only changes the IP address of the packet itself, not the IP addresses and ports in the payload. In the case of H.323, it is the call setup packets that contain the address and port information in the payload.

2. H.323 and SIP, as well as other applications such as FTP and RTSP, use bundled sessions. They exchange address and port parameters within a control session to establish data sessions. NAT cannot determine the inter-dependency of the bundled sessions and assigns unrelated addresses and port numbers to these sessions, which does not work.

3. An IP application (such as IP phone) that attempts to originate a session from an external realm will be able to locate its peer in a private realm only when it knows the externally assigned IP address ahead of time. This is a problem for a traditional dynamic NAT, which only permits sessions to be established in one direction.

4. SIP messages may carry URLs that specify signaling addresses in the Contact, To, and From fields. Once they traverse a NAT, the IP addresses and domain names in the host port portion of the URL may not be valid.

Due to these problems and others which have not been mentioned here VoIP applications need special treatment hen NAT traversal is involved.

1.3 Mobility and VoIP

Over the last decade, we have been experiencing tremendous growth of wireless networks as well as the emergence of all kinds of devices (PDAs, handhelds, digital cellular

phones) with different capabilities and processing power. These technological changes are paving the way for many wireless Internet applications exploiting different traffic types (audio, video, image, web, data). Future networking infrastructure should be capable of providing support for these applications to enable users to achieve continuous connectivity and uninterrupted service of their Internet applications as they move about from one network to another. In mobile computing environments, the goal is to provide continuous connectivity as a mobile host moves from one network to another often referred to as terminal mobility. All the needed reconnection occurs automatically and non-interactively . Terminal mobility can be achieved by exploiting mobile IP which enables nomadic users the convenience of seamless, un-tethered roaming and effective application transparency as a mobile node moves from network to network. Another major requirement for full mobility support is the need of an architecture that enables automatic discovery of the user location which changes with mobility of the user - a feature often referred as personal mobility. New application-layer protocols such as SIP can be used to provide this personal mobility [24].

In this work we have shown SIP signalling can support VoIP applications in case of mobility and when the two terminals are behind NATs. Literature on this subject usually considers the case where the mobile node is not behind a NAT hence our approach is novel in that sense.

In chapter two we introduced VoIP. In chapter three we described the protocols used in VoIP. In chapter four the NAT (Network Address Translation) including solutions to overcome it and the mobility concepts are explained. Chapter five introduces the mobility problem and the details of how SIP signaling is used when mobility occurs in the case where the two nodes are behind NATs. Finally conclusions are given.

CHAPTER 2

Introduction To VoIP

2.1 Fundamentals of Encoding and Voice Packetizing

At the beginning, when the telephone network was predominant, the phone network was based on analog signals to transmit voice over the lines. Analog signals begin deteriorating over distance and need to be strengthened by amplifiers. Nowadays, with the advent of the Internet and the sprouting demand for services to proficiently transmit multimedia applications such as voice and video, new technologies have emerged out of this necessity. It used to be that the use of telephone lines which were once upon a time exclusively for analog transmission had to make changes in order to accommodate the digital world [3].

2.1.1 Codecs

When a person wants to transmit data/voice through the Internet from their home, and they are using a dial-up connection to a local ISP, there will be multiple analog to digital conversions along the way until it reaches the modem of the end user and the analog signal gets converted into digital for the computer to process it. However,

if the technology used is all digital from end-to-end, such as ISDN or fiber optics, then the analog to digital conversions are not necessary.

These analog signals are digitized by a codec, a coder-decoder that outputs a 7 or 8 bit number. The codec takes about 8000 samples of analog sound per second and converts each sample into numeric code [34] . The most common codecs use in their techniques PCM (Pulse Code Modulation) or ADPCM. These two techniques take advantage of the redundant characteristics of the waveform and can compress it around 10 percent more . There are many compression algorithms used in codecs to compress speech/sound; among these there are techniques that compress only a simple version of the source information. These source codecs use less bandwidth and are constructed in a linear format. There are a few of these codecs, Linear Predictive Coding (LPC), Code-Excited Linear Prediction (CELP), Multi-Pulse and Multi-Level Quantization (MP-MLQ), which are widely used as compression techniques.

There are certain measures that can be taken to determine the efficiency of the code that is transmitted over a network; these measurements are called MOS or mean opinion score. The MOS scores are subjective, yet useful, since they are rated by different listeners to choose which is the best quality of sound produced with each codec. If the score drops below 2.68, it means that it is unacceptable. Table 2.1 is a chart that demonstrates a few sample codecs including G.711 and G.723.1 and their corresponding MOS score. As for the G.711 PCM codec that is used in the H.323 Protocol Suite, but not exclusively, it has the highest MOS score but also the highest bit rate.

In H.323 the voice/sound codecs can range anywhere from 5.364 kbps, yet, the one that is used in most applications is G.711 because it uses PCM and it demands more bandwidth due to its 56 to 64 kbps output in addition for the reason that it

Compression Method	Bit Rate [kbps]	Framing Size	MOS Score
G.711 PCM	64	0.125	4.1
G.726 ADPCM	32	0.125	3.9
G.729 CS-ACELP	8	10	3.7
G.723.1 MP-MLQ	6.3	30	3.9

Table 2.1: Sample codecs and their MOS score.

was designed for continuous bit-rate networks [6] . When transmission of IP voice packets takes place through the Internet, G.711 is not the best choice because of the almost guaranteed bandwidth necessary and most Internet ISPs have lower capacity bandwidth. The better codec choice for the Internet would be G.723.1 because it was intended to run on lower bit rates.

The codecs must be well designed and efficient, otherwise, the consequences caused by issues such as delay, will affect the end-user. A delay provoked by a codec (handling delay) can not only cause IP packets to time out and be discarded but also the quality of the voice to be poor.

2.1.2 Packetizing IP

Once that voice/sound has been encoded and compressed by the codecs, subsequently comes the process of packetizing IP packets. Data packets are formed from the compressed data bit stream and need to be first arranged in the payload of the packet making them ready for transmission over the TCP/IP network. The packets consists of about 130 bytes each including the overhead. Once they have been formatted, they are put-through just about every 30 ms. Once the IP packets arrive at the destination, they are converted back to data stream and subsequently, uncompressed and back to their original analog signal[21].

2.2 Performance Issues for Voice over IP and Quality of Service (QoS)

After voice has been encoded and packetized various general issues arise regarding the actual problems and requisites that voice packets need to meet up to in any network regardless of what technologies are used (ATM, Ethernet, ISDN). These are general measurements that must be discussed to better understand the implications for voice packets either transiting through an intranet or the Internet.

2.2.1 Delay and its Implications for Network Performance

One of the principal characteristics that voice packets have to overcome or at least meet specific requisites for an acceptable voice packet is delay. This important factor can determine the success or failure of intelligible voice packets for the end user and the end result quality of voice. Also it is important to keep in mind that voice packets are time sensitive documents which should not suffer long detentions because IP packets are destined to time out at a predetermined point. Since voice packets cannot be recovered, once they time out they are destined to be discarded and the transport protocol that is used, usually UDP, is best effort and will not try to correct or resend the packets. There are two types of delays to be considered. The first one of them is propagation delay which is a delay caused by the properties of light while data is in transit on a line. Propagation delay occurs on both copper and fiber optics. The second and more severe type is called handling delay. This one happens because data have to be processed and passed through numerous devices where things such as bottlenecks or high traffic can hold back the packet until it is fully processed. A delay that is still allowed would be circa 200 ms including the compression processing , however, beyond that time span the packet has a limited time to live and it can at any given second

time out before it can reach its destination. Alternatively, in the other case, the packet would arrive but sound blurry[6][15].

2.2.2 Jitter the way in which it affects voice packets

Another matter that affects voice is jitter. This is caused when there are variations during the transmission of a packet. For example if a packet is expected to arrive at the destination at a predetermined time, but it does not due to a change in the transmission and the packet arrives a millisecond early or late, this would cause the voice stream to sound fragmented [6].

2.2.3 Echo, Rcho cancelation, and Echo Suppression

Echo is another element that must be taken into consideration for voice packets. This is the effect of the caller or the person called, hearing his or her own voice after a short detention. This happens when the signal is being transmitted to its destination and some of the energy is reflected back to the caller. This can be distracting and disruptive in a normal conversation. Some of the solutions are to have echo cancelation which can be a feature incorporated in a gateway that listens for the echo signal and then substrates it. A different technique is echo suppression, which is a device that will recognize speech that is coming from one direction, and it will subtract all of the other signals that are going towards the destination where the packet is headed. However the technique, echo cancelation is the only one that would work in a full-duplex scheme, the latter, echo suppression, can work with a half-duplex by having one person talk and the other one listen [6][34].

2.2.4 Bandwidth and Networks

Bandwidth also plays an important role in voice networks. The task would be to reduce the bandwidth being used by voice packets by employing features like silence suppression. One of the concerns about voice packets sent over a network is that they need to be transmitted continuously, thus use up bandwidth. A conversation between two parties needs to have the packets transmitted to and from without too much delay, otherwise the packets will be dropped or not be intelligible. As mentioned above, one technology is silence suppression that works by noticing the periods of silence in a conversation and stops sending IP packets for that duration of time. Studies have demonstrated that during a full-duplex conversation about 36 to 40 percent of the time the conversation is active, the rest of the time is filled with pauses and silence. One of the counter-effects of voice suppression is that there exists the chance of having the first words in the packet clipped [8].

2.2.5 Packet Loss

Packet loss is another attribute that is measured in VoIP. There are tolerable and unacceptable packet loss rates that make a great difference for the quality of a session. If packet loss averages more than 10 percent then it would not be a tolerable loss rate [11]. Usually packets will be dropped under peak loads and during periods of congestion. Because of voice being time sensitive TCP cannot resend the packets, so in order to compensate for packet loss one of the techniques used is to replay the last packet that arrived fine and to then send the redundant information. The two protocols that shall be discussed later, H.323 and SIP can use TCP to signal but for the transfer of the data stream, UDP is the choice. This is because TCP tries to retransmit lost or corrupted packets [3].

2.2.6 Quality of Service: RSVP and Differentiated Services

Quality of Service or QoS is a set of requisites that a network must meet for the applications to be functional [7]. The definition of a service is the important aspects of packet transmission in one direction across a set of one or more paths within a network. These characteristics may be specified in quantitative or statistical terms of throughput, delay, jitter and/or loss, or may otherwise be specified in terms of some relative priority of access to network resources [37]. These requirements are focused on delay and the bandwidth that a network has. Since an IP packet or a data stream has to pass through (depending on the network) usually a couple of routers along the way to its destination, and each router has to check the IP address, first of where it is coming from and later where it is headed, to forward it on a next hop basis. This hopping from router to router has its consequences. It can provoke delays (handling) along its way [7]. The aspiration of any given network would be to keep high production and reliability in its delivery of packets while not wasting gratuitously bandwidth. One way to meet network ground rules could be by supplementing the TCP/IP based network with RSVP.

2.2.7 Resource Reservation Protocol (RSVP)

This protocol works end-to-end, uninterruptedly, and grants special Quality of Service to applications that may require it, such as sound and video. The Resource Reservation Protocol, RSVP, works on the transport layer above IP. Its main task is to reserve bandwidth so that the audio stream is not delayed or broken up when heading towards its destination. RSVP works by sending a message to the receiver containing the IP address of the sender and receiver. This flow specification includes the delay limits and the rate for the flow. This path message is sent to the receiver by the routers in

the flows path [7]. Once the receiver has the message, the next step is for the receiver to send out the reservation messages to the routers directly in the path to tell them that they are to be expecting an audio/video stream. This message in turn has the IP addresses of the sender and receiver, the flow specifications, policy and admission control (if the node has adequate resources to meet the QoS). If everything works out, the sender collects the reservation request from the receiver and the sender can begin transmitting the data packets. RSVP works for both unicast and multicast receivers. One thing to keep in mind is that RSVP is not a routing protocol but uses routing protocols for a successful transaction [18]. Some of the best protocols for routing are RIP and OSPF, however, they cannot determine which route or next hop would have the most bandwidth available. This is why RSVP is such a valuable protocol for reserving bandwidth along each node along the path used and thus assuring availability for transmission [18][17].

2.2.8 Differentiated Services and QoS

Other methods are currently being discussed in RFC 2474 75 with regards to improving Internet traffic with a concept called Differentiated Services [37][2][18]. Being able to offer different services according to different network architectures and topologies is a great odyssey. Differentiated Services uses a method to mark a packets DS field in the header in order for that packet to receive a particular forwarding handling on a per-hop behavior basis at each network node. The DS field, also known as the code point, consists of six bits in the IP header using the two octets. Using a per-hop behavior would allow to allocate buffer and bandwidth resources at each node in the midst of other competing streams. Differentiated Services is based mostly on the per-hop behaviors and behavior aggregates as their main structure. If Differentiated

Services is implemented on a network, the IP packets would have a certain priority since Differentiated Services works by forwarding according to per-hop behaviors and not just forwarding at random. This is exactly what is needed to have also fewer delays in the transmission of voice and video, since the routing hops are chosen according to how much bandwidth is necessary, also the best route for each IP packet [10].

2.3 Complimentary Protocols that work side by side with H.323 and SIP: RTP/RTCP, RTSP, MGCP/GLP

The main protocols used for voice over IP are the ITUs H.323 and the IETFs Session Initiation Protocol [31]. Both of these work in junction with other protocols to provide different aspects. Internet telephony was first invented to provide point-to-point voice transport between two IP hosts. Due to the development and demand of voice and video over IP, services have had to be added to integrate them with the already existing networks. The services include features such as conferencing, call control, multimedia transport, and mobility and others that must function simultaneously with them, e.g., mail. Since Internet telephony and the traditional circuit-switched networks will not dissipate overnight, equipment such as gateways is needed to do the bridging . In order for SIP and H.323 to work, there is a couple of other protocols that are essential in assisting, transporting the media stream for a end-to-end connection: RTP, RTSP, MGCP/MEGACO and RTCP [32].

2.3.1 RTP and RTCP

The Real Time Transport Protocol is the Internet standard protocol for the transport of real time data, mostly for sound (voice) and video. RTP is formed of a data part and a control part that is knows as the real time control protocol (RTSP). RTP is

used for carrying media streams that are time sensitive [30]. Usually RTP is used with UDP for transporting the data flows however there is no explicit requisite for using any particular lower layer protocol. The way that RTP packetizes media is similar to TCP and IP. It takes the data flow, appends the RTP header and puts the flow in the UDP payload for transmission. The basic function of RTP is to detect packet loss and to re-sequence the incoming stream. The header includes information that is necessary for the receiver to reconstruct the flow (stream). Moreover, the header will have other information that notifies the receiver how the data stream was broken up by the codec.Another function that the transport protocol must be capable of doing is letting the receiver discover packet loss. The RTP header consists of a few fields: V is the version and it is used to identify the session. P is the padding. X is the extension bit. When it is set, the fixed header is followed by exactly one header extension, with a defined format. CSRC count, this contains the number of CSRC identifiers that follow the fixed header. M for marker is used to allow significant events such as frame boundaries to be marked in the packet stream. Payload type, identifies the format of the RTP payload and decides how it will be interpreted by the application. Sequence number, is incremented by one for each RTP data packet sent and may be used by the receiver to detect packet loss and to restore the packet sequence. Timestamp, shows when the first octet was stamped for the RTP data packet. This timestamp must be sampled by a clock that increments perfectly so that the synchronizations are precise. The SSRC identifies the synchronization source. This is an identifier that is randomly chosen, in order to have synchronization sources the SSRC will be the same for the RTP session. CSRC is the contributing source identifiers, which identifies the sources for the payload that is enclosed in the packet.

The RTP properties consist of a variety of fields that are distinct from what other

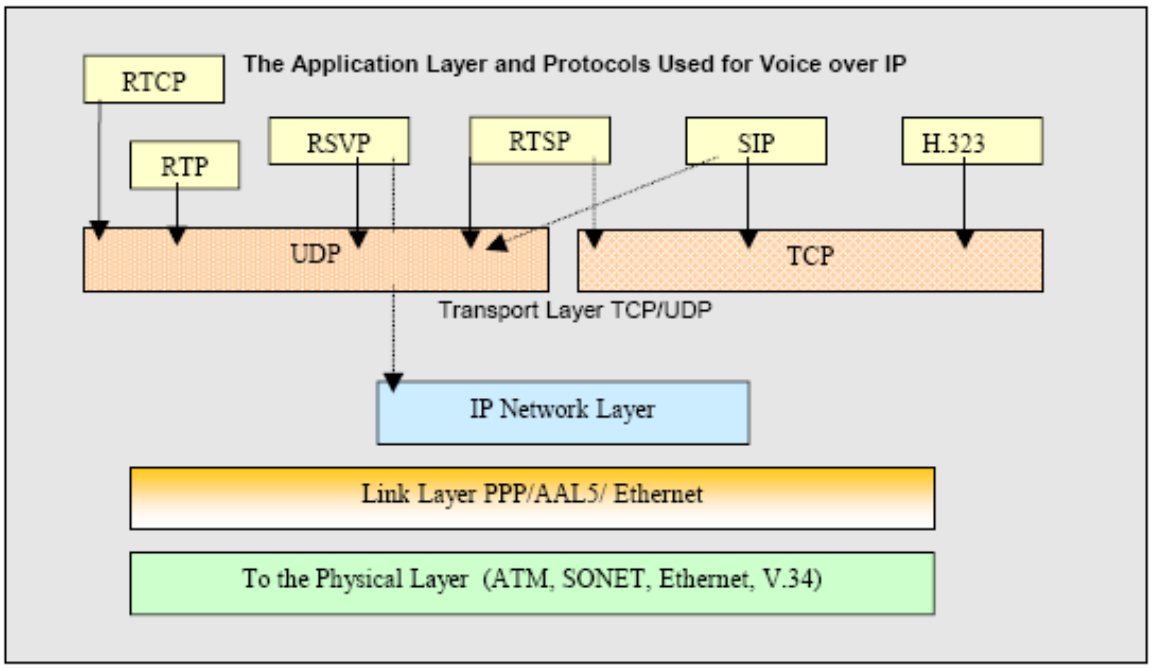

Figure 2.1: Composite of protocols used in Voice over IP and the architecture [31]

protocols use, mainly because this is a real-time protocol. Sequencing is done when the packets arrive out of order. They are put back in sequence at the receiving end based on real time. However, if a packet is lost, once the loss is discovered by the receiver there must be a compensation without having to resend the packet. Intra-media synchronization is defined as the quantity of time that it takes the consecutive packets to be applied or played. Usually when there is silence packets are not transmitted by using the silence suppression technique, so during this period of silence it should be accurate according to the duration of the silence for the receiving end [31].

Inter-media synchronization is employed when diverse types of media are being used simultaneously, and need to be coordinated. An example of having to do lip synchronization would be when someone is watching a video and the person appears to be talking but no sound is emitted. Payload identification is another element that is considered in a session because there can be a variety of changes in bandwidth during the session. When these changes occur, some packets will be encoded using different

codecs. So there needs to be a way of distinguishing which method was used for each packet. Frames are logical elements used to send sound and video and are used by RTP. In order for the receiver to know the beginning and end of a frame, there has to be a mechanism used to delimit them, and facilitate the job of the receiver (similar to flags). There are other functions that make possible to provide more flexibility to the media being transmitted, for example, one aspect of RTP is that it is media independent. It makes available services for real time media, and if a different codec is used, they will be described in the header so that the receiving end is able to decode them. It is also known as multicast-friendly, meaning that this design was devised for multicasting to small or large groups. A certain degree of QoS is doable with the control protocol RTCP due to its feedback property. It can inform the group(s) of how superior the reception is or if they should change their receiving or multicasting data rate (faster or slower). Yet other elements are translators and mixers. The translators are used to take the stream and change its format and decrement the bandwidth if the destination is a network that doesnt have high bandwidth. Mixers work by taking many streams and mixing them together, then outputting a combined result and forwarding the just created stream. One more feature of RTCP is called the loose session control. It is used for enabling different parties to interchange information about each other such as their e-mail address or their telephone number. Encryption is also supported by RTP media streams, however, the users have to exchange their public keys before initiating the session and this can be done by e-mail or anything else that they want to use [13][27].

2.3.2 RTCP

This protocol works hand in hand with RTP. It provides feedback to the participants of a session about the Quality of Service related to the call. There are two main

components for RTCP, one are the Sender Reports that are made by the users that are sending the media and these messages provide information about how much data has been sent, and check out the timestamp on the RTP header to make sure that the data is matched. The other element are the Receiver Reports that are sent by the parties receiving the media. The messages describe how much packet loss has occurred so far in the session, and show the last timestamp and delay since the last sender report was received. There is another feature used mainly for having control during the session. It is called Source Descriptor (SDES) and encloses a field that identifies a particular user. This is through the CNAME domain that is a unique identifier that resembles an e-mail address, and it is used for determining if there are any problems with the SSRC value (RTP header). There are also SDES packets that reveal all the identities of the participants and contain additional contact information. All the reports that are sent back and forth between the senders and receivers have information that can change at any point in time, so in order to have always the updated version, the messages should be sent periodically. This is to keep the entire session smooth and be constantly checking the bandwidth changes and packet loss and to append new participants that want to join the call. Overall, RTP and RTCP are key protocols used for multimedia transport and control and are both used in SIP and H.323. For the future, developers are thinking about using the same report sending method to contact routers directly in order to reserve resources and avoid using RSVP all together. This new protocol is known as SRP or Scalable Reservation Protocol, which would be a simplified version of the very detailed RSVP hopefully accomplishing the same end result by defining a field that alerts routers along the way. SRP is still in working progress [31][4].

2.3.3 RTSP

The Real Time Streaming Protocol (RTSP) is a protocol used to interface to a server for real time media. It is mainly used for controlling the delivery of real-time media streams.

RTSP was designed to work with time-based media, mostly streaming video and sound and where there are applications where time sensitivity is an issue. Since RTSP is time sensitive it was devised to be compatible with many timestamp formats, for instance SMPTE. It was invented to control multicasting and stream transmission. The issue of interoperability of streaming media is hard to understand because it is not a question of understanding just one protocol, but to understand how many protocols interact with each other and how to achieve excellent results. RTSP encoders must be able to store content in files that servers can understand in order for the servers to later be able to stream the contents to the media players and in turn the players must also be able to understand the contents[20].

RTSP is used principally for media on demand and it can be used with few or many participants. It is called the VCR of the Internet due to its properties that make it VCR or CD player style. It does not have the capability to deliver data since it is a control protocol, instead it uses RTP for the actual delivery of the media because RTP can tunnel traffic and it can be set to pass firewalls and other equipment used once the settings have been configured. This does not mean that RTP and RTSP must work in ensemble; they can, but not necessarily, because they were created to work with many protocols. For example if a user wants to view a video that is not live, and supposing that the user had the link to the site, then the transmission would take place directly without RTP. RTSP establishes and controls media streams for a continuous sound or video session between the user and the server. It works in the following way: a user

sends a request to a media server to setup a session and send back the requested data. Another function would be to invite the media server to playback media at a conference or to record a presentation [29]. In addition, an interesting detail is that the server and users can notify each other if new media is to be inserted during the session[25].

The way that RTSP is structured resembles HTTP because its header is text based and uses MIME. The way that the requests and responses are interchanged is similar in the subsequent ways:

1. It needs a request line, headers and a body

2. It uses status codes and security features such as encryption

3. It can also negotiate the contents

4. It has the URL format

A way that it differs from HTTP is in that the server that conducts the media session needs to have its state defined. It also avoids some errors committed by HTTP such as having relative request paths, no extension mechanisms and 8859.1 coding. During the RTSP session, the connection can be via TCP and have the session managed by the unique identifier, or if the session uses UDP then it is very good for multicasting since it has a very low latency rate. RTSP provides some synchronization by using the RTP reports that would be necessary for a virtual presentation. It can be used for distance education, such as our class, for viewing the lecture and retrieving the material that has been previously stored. If the user only wants to retrieve stored media, it is not necessary to invoke SIP or H.323 since RTSP has a method called SETUP and this is sent directly to the server, which means that the media can be sent directly to the client without the use of signaling.

2.3.4 MGCP/GLP

The Media Gateway Control Protocol (MGCP) was designed to convert audio signals on telephone circuits to data carried over the Internet or other packet networks. MGCP is a proposed standard (RFC 2705) . Since IP telephony is complicated they dont have a switch such as the normal phones do and because of this, IP calls need to be processed by gateways on a more independent basis. Since there are two ways to setup calls, either using H.323 or by using SIP, the MGCP was developed to interoperate with both of them. It supports both H.245 and SDP, which have similar functions so the MG (Media Gateway) was designed to support either. MGCP uses the Simple Gateway Control Protocols and the Internet Protocol Device Control. Its main task is to control the gateways from external call control elements or call agents. This telephony Gateway provides conversion operations between audio signals that are employed on telephone circuits and data packets used by the packet switched networks. There are different types of gateways. A trunking gateway would function from an IP network to the telephone network. Another is the residential type that works between a telephony end user and the voice over IP network. Whereas the access gateway has an analog or digital interface using a PBX and would work over an IP Internet network [3]. One of the principal factors of making a call is being able to either know the address of a person or to have the medium to look it up effectively. If the user/host is not sure of the number of the party that he/she wishes to call, then it must be resolved by the system that would be either an end point that has other end point addresses, or if the system is distributed they could use a DNS name lookup with the hostname of a person. The Media Gateways Control Protocol has a process that a calling gateway can receive a PSTN phone number (E.164) from a PSTN node and can then signal the recipients gatekeeper. This would work along with the Gateway Location Protocol that

tries to find out the called partys gateway by exchanging packets that have identity information through their location servers (LS). These two protocols, GLP and MGCP, would help the gatekeeper send messages to the caller about which gateway is in charge for the called party and also give the gatekeeper the number that is to be called and establish a session with the requested endpoint. These protocols are still in working progress and further specifications will come out in the near future[31][15].

CHAPTER 3

VoIP Protocols

The purpose of this chapter is to demonstrate the operational aspects that form part of SIP and H.323. It is also important to illustrate how both protocols perform many of their functions. They are different in some aspects such as how they set up calls, terminate calls, and in the way which supporting protocols are used to enable the actual transfer of media. Signaling protocols are not in charge of the actual media transfer but rely on other protocols such as RTP, RTSP, and RTCP to do that job. There are however two nearly new protocols that were designed hand in hand with SIP: the Session Description Protocol (SDP) and the Session Announcement Protocol (SAP). We shall discuss each of these supporting protocols and how they work with SIP and H.323 and how they conduct connection setups. There are certain guidelines that first need to be followed by any IP signaling protocol, in order for the protocol to be efficient.

1. Any given IP protocol should be able to map and translate the name of the host and the intended recipient and convert this into an IP address. It must also be able to locate the recipient through DNS, system databases or a special server.

2. Signaling should allow feature negotiation with end systems so that they can agree on features such as parameters and encodings. This is necessary since the parameters will vary from connection to connection depending on the end systems and capabilities.

3. Furthermore, any user/host should be able to invite anyone at any point or also have a user in a multi conference be able to hang up without destroying the connection for the rest of the participants. It should also be possible to modify feature settings during the session without interfering with the call[31].

3.1 Session Initiation Protocol (SIP): Standards and Components

The Session Initiation Protocol, which is currently in the standards track [31] , but has also been previously published as a Request for Comments [28], is a text-based protocol that resembles HTTP and SMTP for beginning interactive session with users. These sessions encompass mostly voice, video and chat. SIP was initially created by the Multiparty Multimedia Session Control (MMUSIC) group as well as the Session Description Protocol (SDP), which works in junction with SIP. The developers of SIP keep close contact with other Internet telephony groups such as IPTEL, PSTN, the Internet Internetworking group whose specification is based on SIP[12]. This is all beneficial because the developers of SIP make many considerations for multifaceted characteristics interrelated with old and newer technologies that have not been successfully implemented elsewhere.

The Session Initiation Protocol is a control protocol that is part of the application layer, and it is used to establish, modify and terminate multimedia sessions or calls [33]. The sessions incorporate Internet multimedia conferences (voice and video) and

Internet telephone calls. Some of the functions of SIP are to translate application layer addresses, establish and manage the calls. Since SIP has the capability to invite one or more users for unicast or multicast sessions, and the participants have the additional ability to terminate a session without interrupting an entire conference. SIP also has the means of mapping names and redirecting the calls which permits the use of ISDN and Intelligent Network telephony subscriber services. In addition, it promotes the use of personal mobility or terminal mobility, so that if a user changes workstation, they can still receive calls. Furthermore, Internet telephony gateways that connect PSTN parties can also use SIP to set up calls between them. Its major components are the User Agent, and the network server. The UA is installed in the SIP terminals and contains two components: the User Agent Client (UAC) which takes care of sending the SIP requests and the User Agent Server (UAS), that is in charge of answering the requests. There are a couple of servers that can be used in SIP for example a proxy or redirect server but they don't necessarily form part of the SIP requirements. A regular SIP call does not depend on the servers however they offer many pros, such as a type of directory assistance. The simplest SIP procedure includes an invitation and ack message [31][33].

3.1.1 SIP Addressing

Signaling is an important factor if two or more people want to set up a call and SIP has a number of functions that must be accomplished before the connection can be established. One of them is name translation and user location this must be done in order to learn the IP address of the other party that wants to engage in a session. Normally, the user has the e-mail of the person that he/she wants to talk to, so in order for SIP to work, the e-mail address or whichever type of address is being used must be

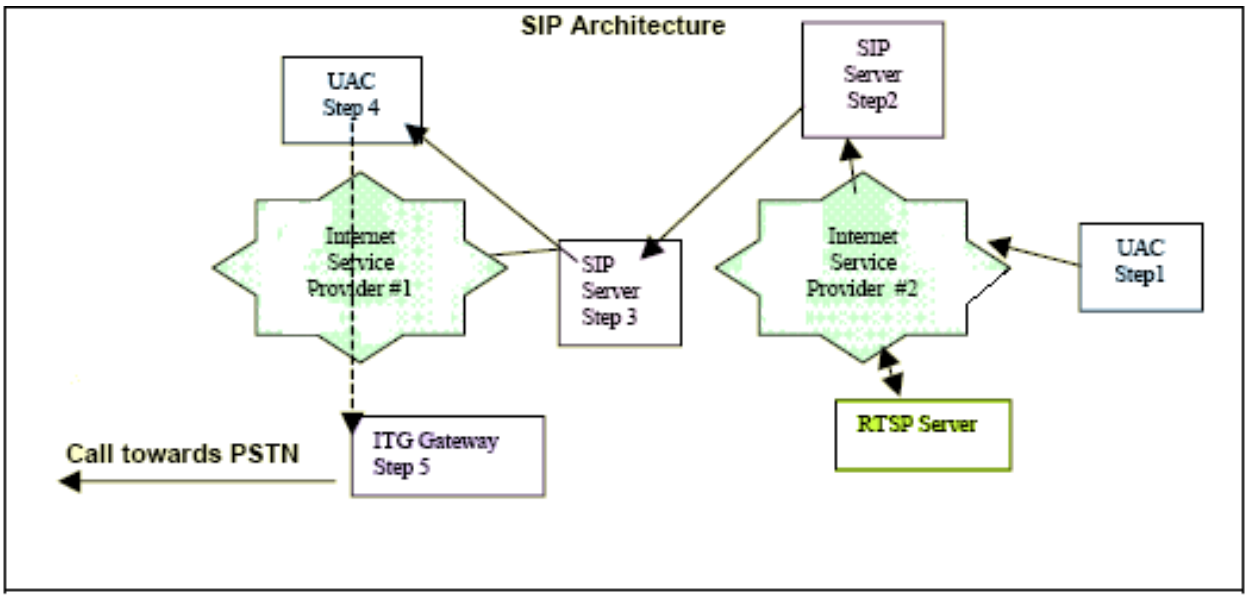

Figure 3.1: General SIP architecture [31].

converted into an IP address. Once an address is resolved it would look something like sip: nazim@ceng.atilim.edu.tr , which looks similar to the telnet or mailto URLs. If the e-mail address of a person were being used, the look up would be simplified because a specific domain has already been assigned and the DNS server would determine the IP faster [31]. This is a very good idea since one can recycle methods instead of creating more addresses and complicating things. Moreover, if a person needed to have authentication, to block unauthorized calls, it can be done since SIP supports this [28].

3.1.2 Locating the SIP Server

Once the host decides that he/she wants to place a call, the user sends a request and the request is sent to a SIP proxy server or if the request has the IP address, it can be sent directly to a port pertaining to the request URI (Uniform Resource Locator). If the request-URI indicates that a particular protocol, mainly TCP or UDP must be used, then the user will contact the server using the designated protocol. If for some reason a protocol is not specified, it will try to reach the server using UDP or TCP and

see which one is accepted for the signaling. At this stage, the user will find one or more addresses for the SIP server by running a DNS query. If the users request-URI is an IP address, then the user can contact the server at the given address. If the request-URI is not an IP address, a DNS server should be queried in order to get the host part of the request-URI [28].

3.1.3 The Unique Call Identifier

The unique call identifier is found in the call-ID field in the header and it must be used by all the participants that take part of the call. Each call has this special number and it serves to not only identify the people, but also for factors such as billing.

3.1.4 Invocation Methods

There are five parts of SIP that are necessary for establishing and terminating a call or multimedia communications application [1]. The first is to establish the user location and decide which end system will be used for the communication to be established. Then the user capabilities need to be specified like which type of media will be used and the parameters need to be defined. Third, comes the user availability that displays the determination of the willingness of the called party to accept the call. The call setup takes place right after the called party has taken the decision that it is ok to accept the call, and at this point the telephone would begin to ring. Last step is call handling, and this includes the transfer and call termination. SIP does not depend on TCP for reliability but instead on SDP for negotiating and identifying which codec(s) will be used for the session [5].

In order to establish calls and connections in SIP there are a few important invocation methods (commands) that are used:

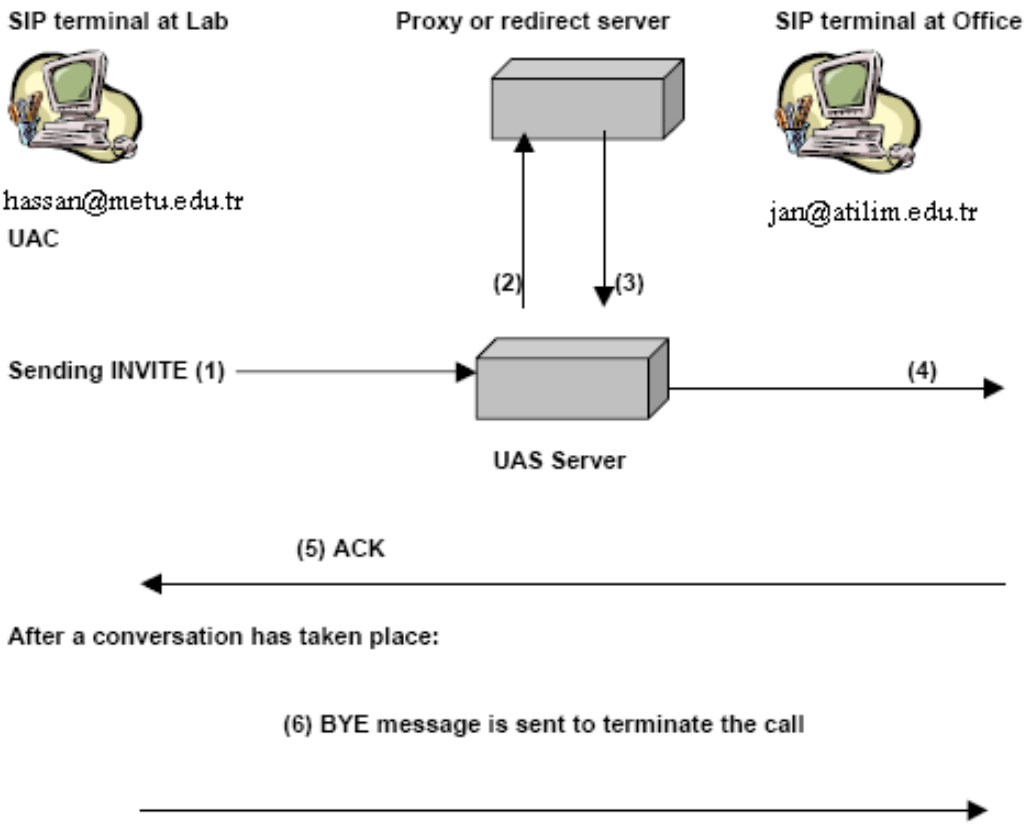

Figure 3.2: SIP transaction.

1. INVITE, just as the name describes, it invites a user to set-up a connection

2. ACK, is used for reliable message exchanges for invitations

3. BYE, releases a connection

4. STATUS, is for servers informing each other on how the signaling is going

5. CANCEL will stop looking for a sought after user

6. REGISTER bestows information about the whereabouts of users to a SIP server

7. OPTIONS this will inspect and plot out the capabilities that each user has. However, it does not have the power to establish a connection

3.1.5 Locating a User After a Change of Location

If a person has changes locations but still wants to be reached, the basic operation would take place. The invite request would be sent to the UAS server and this server would try to contact the user as usual. When a response message is sent back to this server that the user cannot be located, the next step is that the server will forward this message to a redirect server. The redirect server is in charge of keeping forwarding information fro users that have registered their new IP address or location. The information holding the new IP address of the person intended to be contacted by the host is sent back to him/her. The user/originator of the message can now use this to send out the invite message with the new address and get a hold of the intended receiver.

3.1.6 SIP and the Session Description Protocol (SDP)

The Session Description Protocol is used to negotiate capabilities between endpoints in order to establish sessions. For instance in the case that it is needed to exchange codec information for a call by using a text based description. SDP can also transport critical information for RSTSP to negotiate multipoint conference parameters and to ensure that a mutual layout when information of various forms is to be used in SIP [15].

It is used for describing multimedia sessions in order to facilitate the task of the Session Announcement Protocols (SAP), session invitations (SIP) and other types of multimedia commencements. The Session Description Protocol, is used with call agents and it uses MGCP to establish a common communication with the gateways. In addition to codec info exchange it also provides information on the session itself. The name and purpose of the session are described so that the participants can see what type of session it is. It also provides information about when the session is active

or when it is inactive. SDP also informs the participants about aspects such as what type of media is being used in a particular session and even the destination information such as which IP addresses and ports are being used. In addition, it carries information about bandwidth requirements and contact information for the session(s)[16].

3.1.7 SIP and the Session Announcement Protocols (SAP)

Session Announcement Protocol (SAP) is used to assist the advertisement of multicast conferences and other multimedia sessions. These announcements are cast out cyclically and are cached in a directory. To facilitate this a distributed directory is employed and the multicast packets that contain the description of the session are kept there. Other directories can listen to the announcement this way a person or group can listen to the announcement and if they want, they can join the session (usually a conference). The listening party or group can learn from the multicast the scope (the space of opportunity to operate or function) and can notate the SAP address or port for these situations to decide if they want to join a session or not. All of the announcers must listen to all of the other announcements to better determine the total number of sessions being announced in a particular group. By means of the Session Announcement Protocol, there are other factors such as an implicit timeout period that is used in all of the announcements. If a message after being sent out and it has not been received for ten times the announcement period or for one hour, the session will be removed from the receivers cache. A session that has already been announced, but for some reason needs to be modified either during the session itself or before, can be changed by sending a notice in the version field of the announcement header to alert the group(s) that changes have taken place. Then the announcement is sent out by its cyclical multicast and the recipients/ participants will be aware that modifications

for the session have taken place. The announcement has a bandwidth limitation of 4000 bits per announcement and may not exceed that. As for security, encryption of can be performed in the session description by a symmetric algorithm to grant privacy. However, it is not recommended because the announcement would only be targeted to a specific group of key-holders and there would need to be an extra operation for them to have access to an authorized key. In general, the Session Announcement Protocol was largely created to compliment SIP in its operations, but due to its flexibility and independence (being able to interact with other protocols) it would also work well with H.323 [22].

3.2 H.323: The ITU Standards and Protocols Stack

The H.323v2 recommendations from the ITU encompass many different elements and protocols that work together to provide multimedia communications such as Voice over IP, over Local Area Networks. H.323 allows this protocol suite to be integrated in an already existing infrastructure and to promote multimedia applications. The recommendation defines the technical requirements for having audio and video communication services in LANs that do not provide Quality of Service. It does not specify which transport layer is to be used to connect multiple LANs. It does however talk about how H.323 interacts with a Switched Circuit Network (SCN) and all the components that are used in the structure for the communication to take place[6].

There are four major components that form part of H.323 they are the following: terminals/endpoints, gateways, gatekeeper and MCUs. Within all of these components there are specific protocols that work in correlation with each other to setup a connection, establish a session, transport the media stream, terminate the session, and access their own directory services [35]. Since H.323 tries to use all of its own components

and protocols, which were especially, created to work in coordination with one another in order for them to have adequate end results. This however raises many questions with regards to interoperability with non- H.323 application layer protocols such as SDP and SAP that we saw earlier [3].

H.323 within its scope does offer many benefits due to having their own standards. Taking into account that they have their own codec standards for compression and decompression, the best known due to its high quality of 64 bps is G.711, which works quite well according to its MOS score . It extends itself well in matters of interoperability because there will be no doubts about compatibility at the receiving end point, meaning that the recipient of the data stream will be able to decompress or compress data with a well known codec. It is also network independent in that it does not rely on lower layers for additional information and it is also platform and application independent, in other words not tied to any particular hardware in a network. It provides multipoint support for a reasonable amount of points however if the session needs to support more participants it will employ a Multipoint Control Unit for this purpose. In addition, the bandwidth can be managed by an administrator. Multicasting is also supported by H.323 for a moderate amount of participants to have a conference. In addition to multicasting, inter-network conferencing can even be possible because H.323 allows the users to conference remotely [8]. It is also presumed that the transmission path between the users has passed through at least one local area network such as an Ethernet or a token ring [3][15].

3.2.1 Terminals in H.323

Terminals are the physical place where the voice to be sent originates. They are also the place where the multimedia application, either sound or video, will received in the

other side. They are independent devices, or PCs that contain the H.323 protocol stack and include multimedia applications. An endpoint is the point (node) where the user's workstation connects to a LAN to provide real-time and dual communication. Terminals can also communicate with an H.323 gateway or a MCU [3]. The terminals must be able to support H.245 that is used to negotiate channel usage and the capabilities offered. They must sustain Q.931 for call signaling and call setup, as well as a component named RAS that is used to communicate with the gatekeeper. Furthermore, terminals should support RTP and RTCP as their transport medium as well as for sequencing voice and video packets and control of the sessions [11].

3.2.2 H.323 Gateways

H.323 Gateway is a point in the LAN that communicates with the terminals either on the same network or on extended networks. When a terminal is not part of the H.323 type, the gateway translates the transmission formats between the terminals so that they can understand each other for instance if the voice signal codecs are different and the sender uses G.729 while the recipients PC uses G.723.1 [11]. Gateways are important because they provide a connection path from a packet-switched network and the Switched Circuit Network. The gateway may not be necessary if the network doesnt need to connect to other networks and all of the multimedia applications (sound/video) are to be used in one place and need not traverse to a distinct network. Mostly the gateways task is to bounce the media from the Switched Circuit Network to the LANs endpoints [8][9].

3.2.3 H.323 and Gatekeepers

Gatekeepers are another component of H.323. Depending on the type of network, they are not obligatory to have, but if they are present, they need to then perform certain functions. If gatekeepers are used in a network, their most important responsibilities would focus on principally on address translation, admission control, bandwidth control and managing the different zones on a given network. Additionally, they can support other features such as call control signaling, call authorization, bandwidth management, and call management. The endpoints in a network would then register themselves with the gatekeeper. This is essential so if someone wants to place a call, the gatekeeper can get a hold of the address and translate it by consulting a service directory. If a gatekeeper is not present, the gateway would have the responsibility to translate the address or do a look up in a directory. After the gatekeeper has the IP address, it either grants or declines permission to place the call. It is also imperative in the task that the gateway is registered as well as the endpoints so that it can route the media efficiently because it is also in charge of different zones within a network. There are also other duties that are defined, for instance, authentication and authorization of calls, if a endpoint is not registered with the gatekeeper or, if it is specified that some numbers are not authorized for access, they can be blocked. For network administration is very practical to have a gatekeeper to keep track of registered terminals, the number of active terminals, to distribute bandwidth in different zones, or reserve bandwidth for available allocation [11][33][9].

3.2.4 Multipoint Control Units

Multipoint Control Units or MCUs are used when a call or conference needs to keep multiple connections active. Since there could be a moderate number of simultaneous

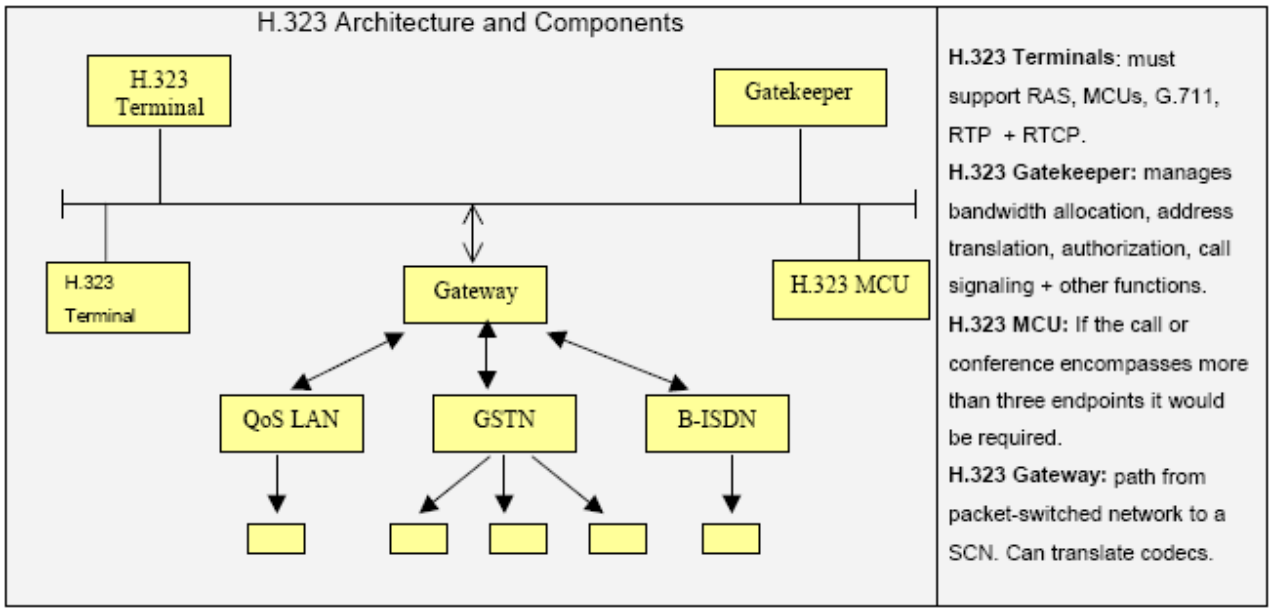

Figure 3.3: H.323 architecture and principal components. The description of each of the components functions are on the right hand column.

connections, these MCUs take care of negotiating between terminals and checking with all of the terminals what capabilities they can dispose of for the conference or call. The MCUs are composed of two parts; a multipoint controller (MC) that is in charge of negotiating and determining each terminals capacity. On the other hand a multipoint processor (MP), is used to process the multimedia, streams during a conference or a multi-point call. For this they need to have anywhere from zero to more MPs because they are in charge of mixing and switching the media stream and also the processing of the audio and video data bits. MC does not interact directly with the media streams, it is actually MPs job to do that. MCs and MPs can either be installed as independent devices or be part of other H.323 components. For multipoint conferences they can either be centralized (all in one network) or decentralized which can be multicast and MCU is not used. For decentralized conferences terminals would use H.245 signaling control to tell the Multipoint Control how many streams they can handle in encoding and decoding and it would be managed in a point-to-point basis [6][15].

3.2.5 Signaling Channels in H.323

Channels are used to send information back and forth from endpoints to gateways, or the gatekeeper to see if a call can be placed. Different channels are used with different methods (messages) to gather necessary information and make sure that terminals can handle connections by exchanging capabilities, and checking what kind of codecs they may use, or if there is sufficient bandwidth to set up the call [6].

This information is passed via channels that have diverse functions, for example, one protocol has a certain task, of contacting a particular device and making an agreement over those important elements that need to be concrete before any connection can be established. A user (terminal) cannot simply put voice packets over IP and expect that they alone reach the recipients end. The combination of components need to work as a team. One example would be when the gatekeeper is responsible for managing finances, or has access to directory services to find out the IP address for a media stream in order for the packets to be able to be put through Ethernet and at pass through the Switched Circuit Network. In order for the entirety of the process to work all the components have to work in synchronicity [11][3][6].

3.2.6 H.245 for Media Control in H.323

H.245 is used to control media at the endpoints by negotiating which capabilities each is able to handle. It works by sending control messages that are interchanged in order to see what each side can provide. During the negotiations H.323 lets endpoints have different sending and receiving potentials. For example, one endpoint can use a different codec or bit rate. The use of messages that provide information about flow control between endpoints or the closing and opening of logical channels is done with H.245 media control. Once the participants have exchanged their means, the next step

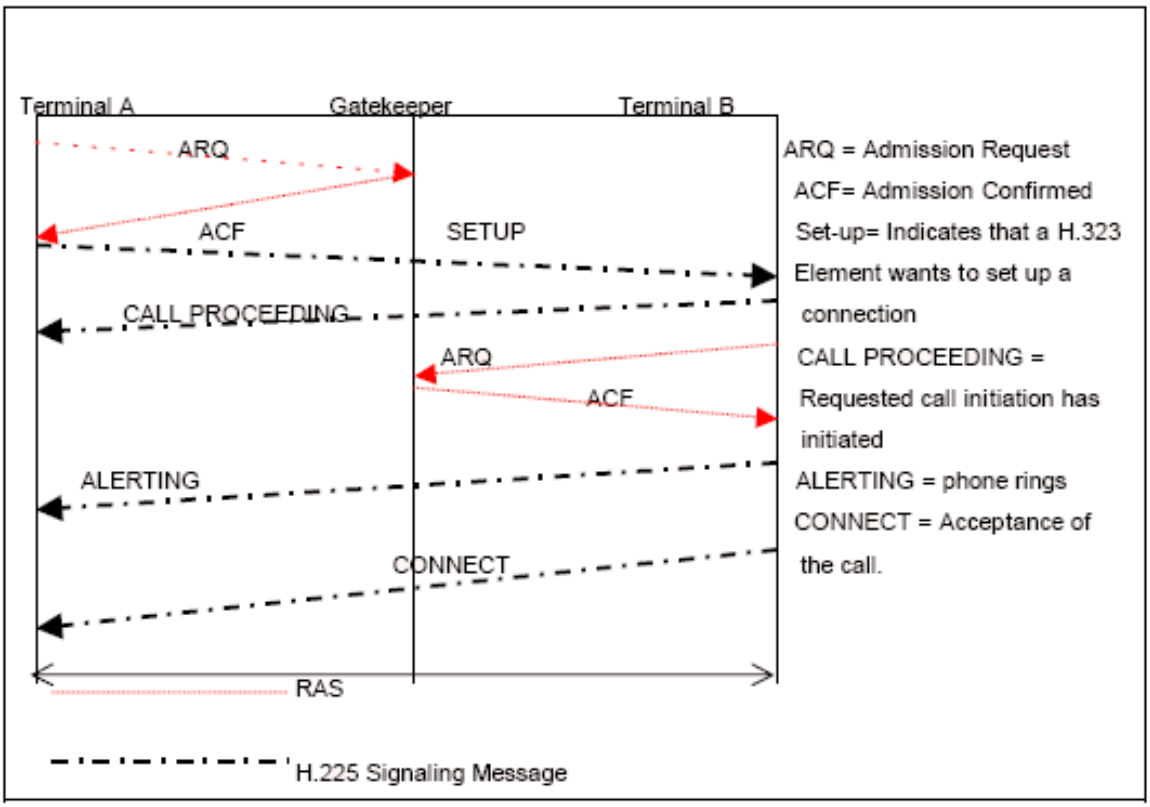

Figure 3.4: RAS and H.225.0.

is to open up a logical channel such as RTP and RTCP via the H.245 control channel in order for the media to be transported from one extreme (endpoint) to the other. The H.245 control protocol is comparable to the session description protocol function wise [9].

3.3 H.225.0: Signaling and RAS

H.225.0 is the elementary process for setting up and terminating calls. H.225.0 is a derivation of Q.931protocol, and it uses a similar message layout for its call signaling. In the case that a gatekeeper should not be present, H.225.0 will send the messages directly from endpoint to endpoint. If there is a gatekeeper the messages can be routed through it. It is also known as the call signaling channel. H.225.0 is needed to transport data that is used for call control and other control assistance. The moment that a call has been set-up, the H.245 transport address will show in this channel. Precise

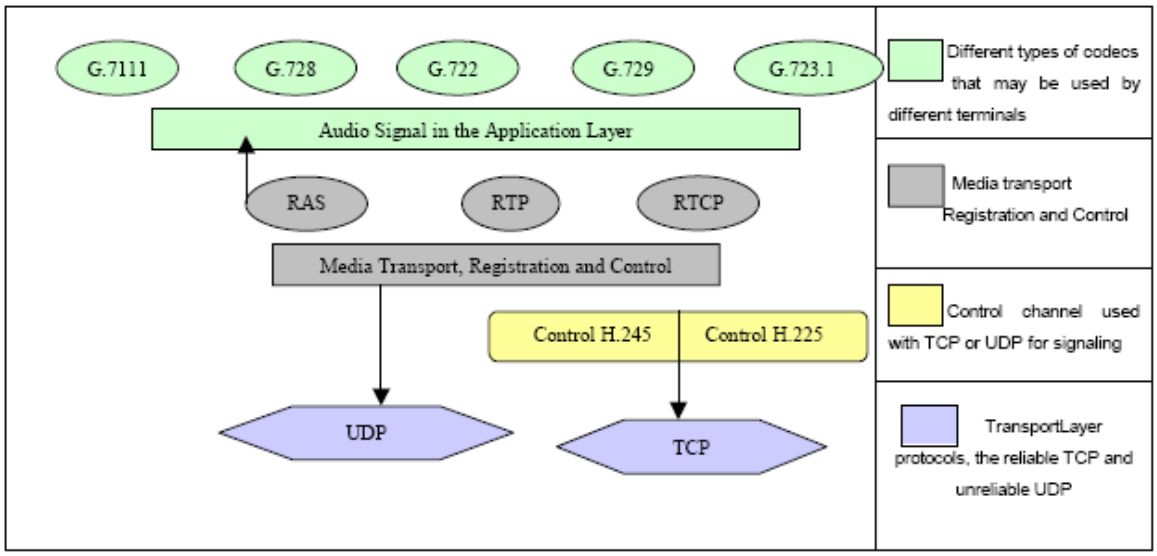

Figure 3.5: Composite of codecs, protocols, and control protocols in H.323.

methods are used to start bi-directional communication [23]. The RAS protocol, has a special procedure that helps to establish communication between an endpoint and the gatekeeper that manages an endpoint. It is known as the Registration, Admission and Status protocol. The Registration method is used so that endpoints can register with the gatekeeper their alias address and later be matched up with their call signaling channel transport address. The RAS channel is also used to exchange RAS messages. This channel has to be opened before the establishment of other channels amid the endpoints and gatekeeper. Once the terminals and gateways have registered with the gatekeeper that is also a zone manager, it will know where to route the media. Admission Procedures are managed by Q.931 that acts like a signaling protocol. The originating user sends an ARQ request to the gatekeeper to say, I would like to place a call, give me consent. The gatekeeper returns a ACF/ARJ (admission request or reject) depending on the bandwidth and other conditions related to RAS.

3.4 SIP and H.323 Compared

There are many differences between SIP and H.323 that are worth examining because these key issues define how fine these protocols are in terms of their interoperability, flexibility, extensibility, and scalability.

SIP is a much simpler protocol since it uses few headers and it depends on higher layer protocols to do other jobs such as the session description protocol to describe and negotiate capabilities during a session. While H.323 defines an almost infinite amount of components and protocols that have to form part of the recommendation making it incompatible with components in systems that do not support H.323. There are differences in the voice setup delay SIP can either use TCP or UDP for its signaling while H.323 exclusively uses TCP for its setup and this unfortunately has the fame of causing more delays. In terms of complexity, SIP is very similar to the text based HTTP protocol that is very efficient. On the other hand H.323 uses a variety of protocols such as H.245 for transporting messages, ASN, H.225.0 and H.450, making it highly complex. As for extensibility, SIP is a protocol that is highly extensible since it was created to operate with basically any existing or future protocols, whereas H.323 it falls short of this because it works with vendor specific ASN.1 which lacks negotiation capabilities with other products and protocols that could be used in a different network. SIP works with any codecs that are registered with IANA, and H.323 uses its own ITU created codecs [31][33].

Their architecture is different in that SIP has a modular structure that includes call signaling, user location and registration, as well as some QoS, session description and announcement protocols. As for H.323, it has a colossal structure that offers many services provided by its own components such as capability exchange, conference control, signaling, QoS, registration and service discovery. Addressing in SIP is quite

simple, involving having a e-mail address, a H.323, http, or any type of URL will do. In H.323 addressing can be resolved using the gatekeeper and for connections to the PSTN (or back to the network) the translation of PSTN E.164 numbers is plausible.

For Transport, SIP can use any protocol that allows the use of UDP in order to save time with the setup connection. H.323 must use TCP or any type of reliable protocol, which can slow down the setup time. Another difference is in the use of how their servers operate. With SIP the servers keep a stateless status which means that after the signaling and setup have been completed, they forget about the call. The servers in H.323 are called stateful because they are kept on call state for the entire duration of the call, even through the TCP states which makes it have lower reliability and scalability (they tie up the servers). SIP uses a distributed multicasting feature while H.323 usually has the central MC that has some problems with large scale conferences creating bottlenecks and for its signaling it can only be done by unicasting. Overall, H.323 defines too many elements, thus adding to the complexity and making it hard to be extendible with other protocols that are not ITU related. Whereas SIP is very compatible and can work with any protocol, even protocols that have not yet been invented. SIP is simpler in its message encoding since it uses text which helps to simplify debugging and H.323 is based on binary representation of messages that are part of ASN.1 and also uses PER which also need a special code generator to parse [15].

SIP is a much simpler and amiable protocol that was designed to work with a great variety of other protocols regardless of where they were created. It has great potential for large conferences with its distributed multicasting capabilities (highly scalable) and most compatible.H.323 has many excellent features as well, however, since it is so complex, it is hard for it to scale conference-wise to anything larger than

a moderate sized conference using MCUs. Another inconvenience is that it duplicates many functions that could be combined in one protocol for example it should combine H.245 and H.225.0 into one to save a couple of round trips that are delay causing. It also uses RTCP and H.245 which both give feedback.

CHAPTER 4

Concepts of Network Address Translation (NAT) and Mobility

4.1 Network Address Translation (NAT)

Network Address Translation (NAT)is being used by many service providers and private individuals as a way to get around the problem of not having enough IP addresses. An enterprise may have a block of IP addresses assigned to them, but many more computers than the allocated IP addresses.Alternatively,an individual may have a DSL connection with one IP address,but want to have multiple computers hooked up to the Internet. NAT solves this problem by mapping internal addresses to external or public addresses.An internal IP address:port pair is mapped to an external IP:port,and whenever the NAT receives a packet with the external IP:port,it knows how to reroute the packet back to the internal IP address and port.The mapping is valid for some predefined mapping interval after which,in the absence of network traffic between the two communicating parties,this mapping may be expunged.In all cases,it is assume that an application will send and receive packets on the same port Fig.4.1.

There are four types of NATs. As defined in [19] they are:

1. Full Cone

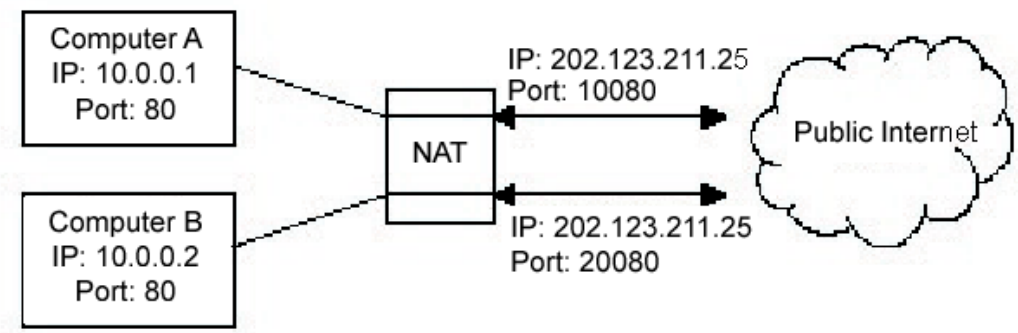

Figure 4.1: NAT schematic

2. Restricted Cone
3. Port Restricted Cone
4. Symmetric

For a given internal address,the first three types of NAT maintain a mapping of this internal address that is independent of the destination address being sought.The fourth type of NAT will allocate a new mapping for each independent destination address.

Unless the NAT has a static mapping table,the mapping that opens when the first packet is sent out from a client through the NAT may only be valid for a certain amount of time (typically a few minutes),unless packets continue to be sent and received on that IP:port.

4.1.1 Types of NATs

4.1.1.1 Full Cone NAT

In the case of the full cone,the mapping is well established and anyone from the public Internet that wants to reach a client behind a NAT,needs only to know the mapping scheme in order to send packets to it.

For example,a computer behind a NAT with IP 10.0.0.1 sending and receiving on port 8000,is mapped to the external IP:port on the NAT of 202.123.211.25:12345.

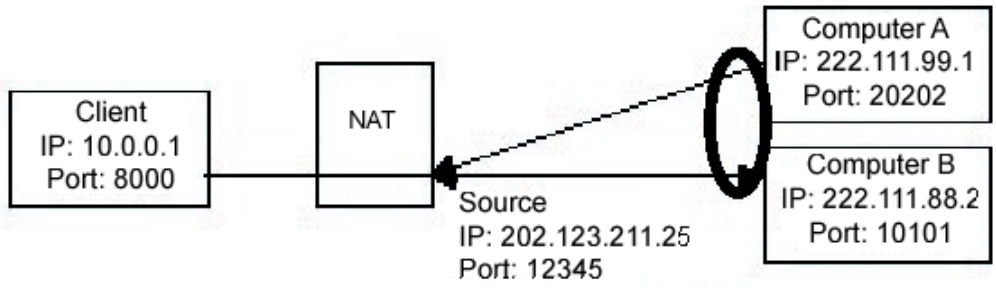

Figure 4.2: Full Cone NAT

Anyone on the Internet can send packets to that IP:port and those packets will be passed on to the client machine listening on 10.0.0.1:8000 Fig.4.2.

4.1.1.2 Restricted Cone NAT

In the case of a restricted cone NAT,the external IP:port pair is only opened up once the internal computer sends out data to a specific destination IP.For example,in the case where the client sends out a packet to external computer 1,the NAT maps the client s 10.0.0.1:8000 to 202.123.211.25:12345,and External 1 can send back packets to that destination.However,the NAT will block packets coming from External 2,until the client sends out a packet to External 2 s IP address.Once that is done,both External 1 and External 2 can send packets back to the client,and they will both have the same mapping through the NAT.

4.1.1.3 Port Restricted Cone NAT

It is almost identical to a restricted cone,but in this case the NAT will block all packets unless the client had previously sent out a packet to the IP AND port that is sending to the NAT.So if the client sends to External 1 to port 10101,the NAT will only allow

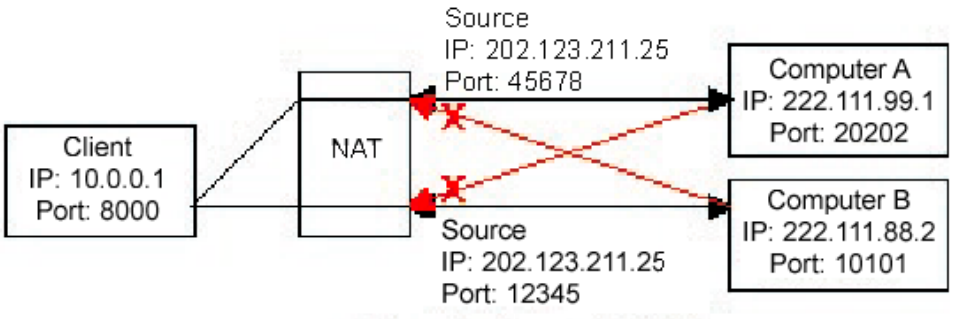

Figure 4.3: Symmetric NAT

through packets to the client that come from 222.111.88.2:10101.Again,if the client has sent out packets to multiple IP:port pairs,they can all respond to the client,and all of them will respond to the same mapped IP:port on the NAT.

4.1.1.4 Symmetric NAT

The last type of NATs is different from the first three in that a specific mapping of internal IP:port to the NATs public IP:port is dependant on the destination IP address that the packet is sent to.So for example,if the client sends from 10.0.0.1:8000 to Computer B,it may be mapped as 202.123.211.25:12345,whereas if the client sends from the same port (10.0.0.1:8000)to a different IP,it is mapped differently (202.123.211.25:45678)Fig.4.3.

Computer B can only respond to it s mapping and Computer A can only respond to it s mapping. If either one tries to send to the other s mapped IP:port,those packets will be dropped.As in the case of the restricted NAT,the external IP:port pair is only opened up once the internal computer sends out data to a specific destination.

4.1.2 Possible Solutions for NAT Traversal

If the client is behind one of the first three NAT types,then the solution for NAT traversal is fairly simple.The client must find out how its internal IP:port looks to

the world (i.e.the NAT mapping) and then it must put that information into the SDP message instead of the information reflecting its internal IP:port.There are two methods for a client to determine the NAT mapped public IP:port.The first is to ask the NAT,the second is to ask someone outside the NAT on the public Internet.

4.1.2.1 UPnP

A client can ask the NAT how it would map a particular IP:port through a protocol called Universal Plug and Play (UPnP).This is a solution that is being pushed by Microsoft (among others).The client queries the NAT via UPnP asking what mapping it should use if it wants to receive on port x.The NAT responds with the IP:port pair that someone on the public Internet should use to reach the client on that port.Many NAT device manufacturers have already included UPnP in their products.

One problem with UPnP is that it will not work in the case of cascading NATs.For example,say an ISP owns a block of IP addresses,but not enough to service its user base.The ISP would use a NAT to provide IP addresses to its customers.One of those customers may require many IP addresses (for example,an Internet caf) so it would set up its own NAT to share its one address between many computers.If a client running on one of the local computers were to use UPnP to determine its public IP:port,then it would only get back the innermost mapping (that of the Internet caf s NAT)but would still have a one way voice problem.That is because the public Internet would still not recognize the IP:port that the client was giving,since a second translation occurs between the Internet caf s NAT and the public Internet via the ISPs NAT.There are also security issues that have not yet been addressed with UPnP. Furthermore,there is a huge installed base of existing NATs that do not support UPnP.It is not realistic to retrofit this installed base in the near future.

4.1.2.2 STUN

Simple Traversal of UDP Through NATs (STUN) is a protocol for setting up the kind of NAT Probe that was just described.It actually does a bit more than just return the public IP:port it can also help determine which kind of NAT you are behind.Clients are already being developed that are STUN aware and can set their SDP messages accordingly. STUN requests specify the following parameters:

- RESPONSE-ADDRESS.
- Change IP.
- Change Port.

The STUN server will send its response to the IP:port specified in the RESPONSE-ADDRESS if that field is not present,then the server sends its response to the IP:port that it received the request from. If both the Change IP and Change Port flags are not set,the STUN server responds from the IP:port that the initial packet was sent to. If the change IP flag is set,the server replies from a different IP,and if the Change Port flag is set,the server replies from a different port.

The STUN response contains the following information:

- MAPPED-ADDRESS the IP:port of the client as seen by the first STUN server outside the NAT to receive the STUN request.
- CHANGED-ADDRESS -the IP address that would be the source of the returned response if the request had the change IP fl ag set.
- SOURCE-ADDRESS the IP::port where the STUN response was sent from.

Using a combination of different requests to a STUN server,a client can determine:

- If it is on the open Internet.
- If it is behind a firewall that blocks UDP.
- If it is behind a NAT,and what type of NAT it is behind.

4.2 Mobility

4.2.1 Mobility Terms

There are common terms that are used. They are

a. *IP Addresses*

Commonly the IP address usually represents the location of the computer. To solve this problem for mobility, two IP addresses are used in mobile equipment: one constant address for the endpoint identifer, and one temporary address known as care-of-address giving the location of the mobile.

b. *Mobility Agents*

There are two kinds of mobility agents: the home agent and the foreign agent, which keep track of the mobile's location. Each mobile firstly registers in the home agent in its home network, and is administrated by the foreign agent when visiting an the foreign network. In the latter case, the mobile has to obtain the care-of-address and that foreign agent cooperates with the home agent to deliver the datagram to the mobile.

c. *Routing Schemes*

There are two routing schemes for mobile IP. The first one is tunneling, that is, the home agent attracts all packets sent to the mobile and encapsulates them using IP-within-IP en- capsulation, finally sends the resulting packets to the care-of-address of the mobile. An- other scheme is route optimization. It enables to send packets directly to the care-of - address of the mobile, depending on an updated mobility binding provided to all peer communicating with the mobile node. Route optimization can offer smoother handoff than tunneling .

4.2.2 Modes of Mobility

a. *Terminal Mobility*

"Terminal mobility allows a device to move between IP subnets, while continuing to be reachable for incoming requests and maintaining sessions across subnet changes". SIP location management server with additional software is used for the location management of terminal mobility. In the location server, the user name, terminal identifier and terminal location are stored. Terminal mobility in SIP involves three stages: pre-call, mid- call and to recover from network. In addition, Hierarchical registration and Handover performance, RTP, TCP-based application and streaming multimedia applications are sub-issues to be discussed for terminal mobility in .

b. *Session mobility*

Session mobility allows a user to maintain a media session when changing terminals. In order to implement the session mobility using SIP, the primary end system A configures the other end system B, which is to receive and send the media stream. And then A conveys its IP addresses and ports to B using a new INVITE request. Two solutions can be used as configuration mechanisms. One is the third-party call control, the other is the REFER mechanism, depicted more in .

c. *Personal mobility*

Personal mobility allows addressing a single user located at different terminals by the same logical address [11]. In the other words, one address to many potential terminals mapping and many addresses to one terminal mapping are allowed. SIP forking proxies [2] are used to make the device choice transparent to third parties,

then a user can be reached at any of devices via the same name. Moreover, the registrars may recognize which different devices belong to the same person by using a number of heuristics such as chen.zhang@cc.hut.fi, cxzhang@cc.hut.fi, and doris.zhang@cc.hut.fi, which are all part of the same logical entity.

d. *Service mobility*

Service mobility allows the user to maintain access to their services when moving or changing devices and network service providers. Service mobility in SIP requires the SIP application to register with the registrar periodically (one hour) or whenever the network address changes. And the registration conveys parameter information to the registrar including the current network address, properties of the device and user configuration elements. The UA also uploads its time stamped version of the configuration information. The server updates its own version or returns a more recent copy in the registration response .

4.3 Mobility Using SIP

SIP supports personal mobility, i.e., a user can be found independent of location and network device (PC, laptop, IP phone, etc.).

But user can change location (IP address) during a traffic flow, this is called IP mobility. In order to support IP mobility, we need to add the ability to move while a session is active. It is assumed that the mobile host belongs to a home network, on which there is a SIP server , which receives registrations from the mobile host each time it changes location.

This is similar to home agent registration in Mobile IP. Note that the mobile host does not need to have a statically allocated IP address on the home network. If the mobile host moves during a session, it must send a new INVITE to the correspondent

host using the same call identifier as in the original call setup. It should put the new IP address in the Contact field of the SIP message, which tells the correspondent host where it wants to receive future SIP messages. To redirect the data traffic flow, it indicates the new address in the SDP field, where it specifies transport address all this is explained in the following proposal.

CHAPTER 5

Proposed Solution Using Pure SIP Messaging to NAT Traverse in Case of Mobility

We explain our work as a scenario of two nodes after been introduced to each other. One of them moved to a new location as shown in Figure 5.1.

First Ali picks his VoIP phone and dials the SIP address Jan@metu.edu.tr that is the UA (Ali) sends an INVITE-request (1) to the outbound proxy for delivery to Jan, recording IP to the Via-header field, globally unique SIP session identifier to the Call-ID -header field as well as direct contact information to the Contact-header field. To assist proxies, the UA adds the tag identifier, in order to allow Jan's UA to eventually finalize the dialogue by adding another tag:

```
INVITE sip:jan@metu.edu.tr SIP/2.0
Via: SIP/2.0/UDP vo1.hq.atilim.edu.tr,rport
To: Jan<sip:jan@metu.edu.tr>
From: Ali <sip:ali@atilim.edu.tr>;
tag=18271
```

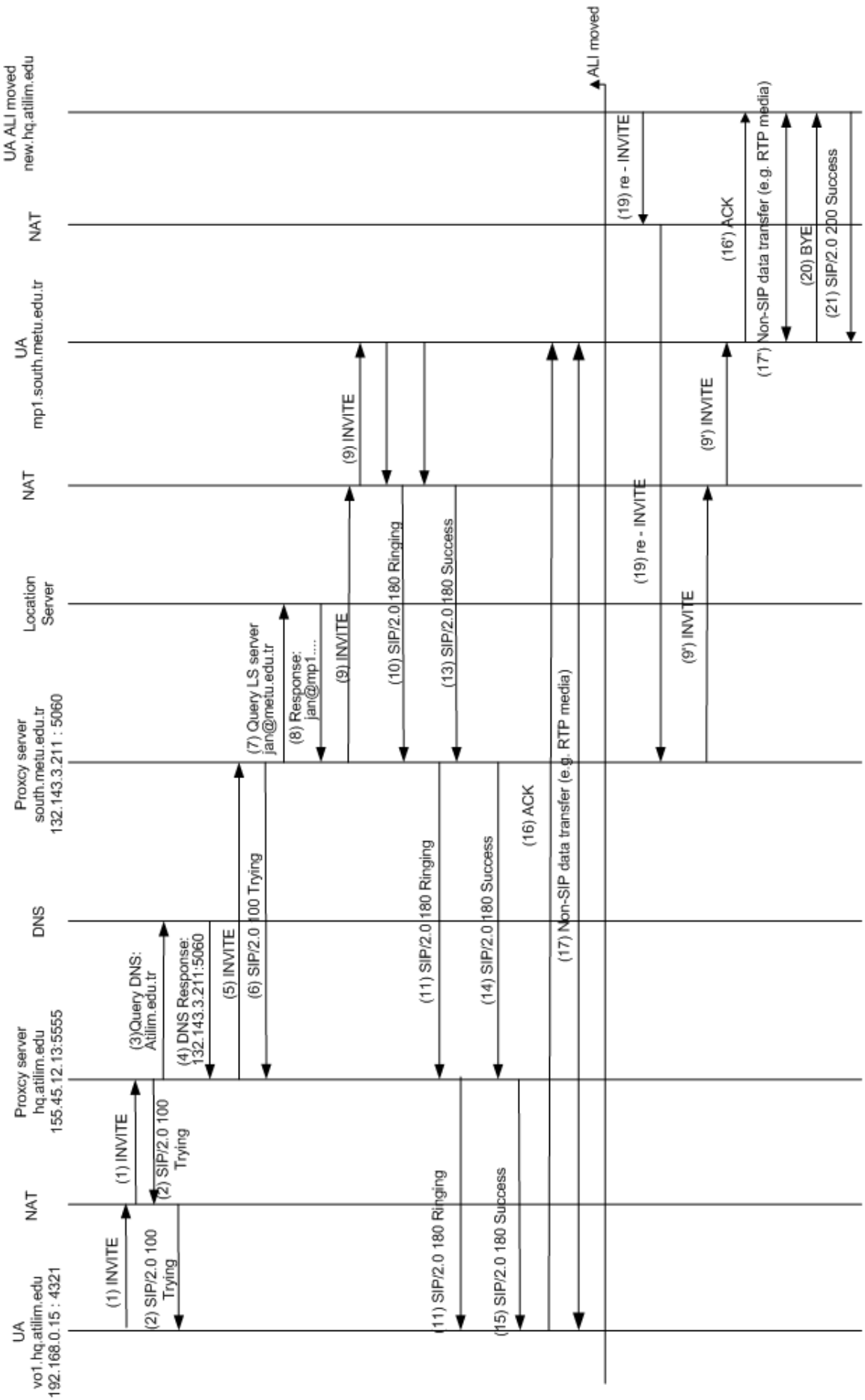

Figure 5.1: The proposed flow of signalling

```
Call-ID: 3223842@vo1.hq.atilim.edu.tr

CSeq: 12921 INVITE

Contact: <sip:ali@vo1.hq.atilim.edu.tr>
```

The outbound proxy, once receiving this, replies with a provisional response, indicating that the specific session is being processed (2):

```
SIP/2.0 100 Trying

To: Jan<sip:jan@metu.edu.tr>,rport=8050

From: Ali <sip:ali@atilim.edu.tr>;

tag=18271

CSeq: 12921 INVITE

Call-ID: 3223842@vo1.hq.atilim.edu.tr
```

In addition to this, the outbound proxy does a DNS query (3) for the metu.edu.tr sip service and receives Jan's inbound proxy-server IP address, protocol and port as a reply (4), as there are no stateless intermediary proxies. The reply indicates that e.g. UDP is preferred, but TCP is also available. IP address 132.143.3.211 prefers UDP and has the service in port 5060. Ali's outbound proxy forwards the request (5), adding only its own Via-header field

```
INVITE sip:jan@south.metu.edu.tr SIP/2.0

Via: SIP/2.0/UDP hq.atilim.edu.tr

Via: SIP/2.0/UDP vo1.hq.atilim.edu.tr

To: Jan<sip:jan@metu.edu.tr>

From: Ali <sip:ali@atilim.edu.tr>;

tag=18271

Call-ID: 3223842@ vo1.hq.atilim.edu.tr
```

```
CSeq: 12921 INVITE

Contact: <sip:ali@vo1.hq.atilim.edu.tr>
```

Jan's inbound proxy receives this and sends a provisional response (6), similar to (2) but the Ali's outbound proxy adds a Via-header field. The outbound proxy decides not to send this to Ali's UAC since it has already sent this to it. The inbound proxy first priority is locating Jan's contact address. It therefore queries the server providing location service for Jan's contact address (7). The location server responds (8) with the IP address (either only address or preferred address that Jan defined through a registrar), protocol and port of mp1.south.metu.edu.tr. Jan's inbound proxy adds its Via-header field, rewrites the INVITE SIP URI and forwards the request (9):

```
INVITE sip:jan@mp1.metu.edu.tr

SIP/2.0,rport

Via: SIP/2.0/UDP south.metu.edu.tr

Via: SIP/2.0/UDP hq.atilim.edu.tr

Via: SIP/2.0/UDP vo1.hq.atilim.edu.tr

To: Jan<sip:jan@metu.edu.tr>

From: Ali <sip:ali@atilim.edu.tr>;

tag=18271

Call-ID: 3223842@vo1.hq.atilim.edu.tr

CSeq: 12921 INVITE

Contact: <sip:ali@vo1.hq.atilim.edu.tr>
```

As the request arrives to Jan's VoIP phone, it starts ringing. The phone sends a provisional response. This now includes the locally unique tag in the To-header field, establishing an early dialogue, prior to session being established and direct contact address to Jan(10):

```
SIP/2.0 180 Ringing,rport= 8050
Via: SIP/2.0/UDP south.metu.edu.tr
Via: SIP/2.0/UDP hq.atilim.edu.tr
Via: SIP/2.0/UDP vo1.hq.atilim.edu.tr
To: Jan<sip:jan@metu.edu.tr>; tag=129991
From: Ali <ali@atilim.edu.tr>; tag=18271
Call-ID: 3223842@vo1.hq.atilim.edu.tr
CSeq: 12921 INVITE
Contact: <sip:jan@mp1.south.metu.edu.tr>
```

The response is sent (11) by inbound proxy to Ali's outbound proxy with one Via-header field (Jan's proxy) removed and onward (12) to Ali's UA with still one more Via-header field (Ali's proxy server) removed. While the provisional response is being routed on the application layer, Jan picks up the phone. Jan's phone sends a final response (13) to Jan's proxy as a notification of success:

```
SIP/2.0 200 Success
Via: SIP/2.0/UDP south.metu.edu.tr
Via: SIP/2.0/UDP hq.atilim.edu.tr
Via: SIP/2.0/UDP vo1.hq.atilim.edu.tr
To: Jan<sip:jan@metu.edu.tr>; tag=129991
From: Ali <ali@atilim.edu.tr>; tag=18271
Call-ID: 3223842@vo1.hq.atilim.edu.tr
CSeq: 12921 INVITE
Contact: <sip:jan@mp1.south.metu.edu.tr>
```

The response is forwarded (14) by Jan's outbound proxy to Ali's outbound proxy with one Via-header field (Jan's proxy) removed and onward (15) to to Ali's UA with still

one more Via-header field (Ali's proxy server) to remove.

To provide reliability, Ali's phone acknowledges the final response. This is sent directly to Jan's UA (16):

```
ACK sip:jan@mp1.south.metu.edu.tr SIP/2.0
Via: SIP/2.0/UDP vo1.hq.atilim.edu.tr
To: Jan<sip:jan@metu.edu.tr>; tag=129991
From: Ali <ali@atilim.edu.tr>; tag=18271
Call-ID: 3223842@vo1.hq.atilim.edu.tr
CSeq: 12922 ACK
```

After this, the media session begins (17).

The effects of user mobility would depend on how it is done. In case Jan was in the move with a Mobile IP-based terminal, the tunnel from Home Agent to Foreign agent and from there to Jan's terminal would be responsible for the session maintenance and basic routing would carry the message from Jan's terminal to the in-bound proxy in Jan's home network. SIP would be totally oblivious to this as it is done in the network layer. Ali's UAC would assume the Home Agent address is Jan's terminal. On the other hand, in case Jan was e.g. visiting a daughter company of metu.edu.tr and had a possibility to register there, he would have used the Registrar to change metu.edu.tr location service to redirect all calls to his SIP address to a new SIP address, e.g. Jan@minibus.edu.tr. In this case Ali's SIP INVITE would have proceeded as in Figure 5.1 up until the locations server response (8), which would indicate the new address. The proxy would generate the redirect response, also indicating how long the stateful proxies and UA can cache the information (for new session establishing purposes, prior to renewing the information:

```
SIP/2.0 302 Moved Temporarily
```

```
Via: SIP/2.0/UDP hq.atilim.edu.tr
Via: SIP/2.0/UDP new.hq.atilim.edu.tr
To: jan@metu.edu.tr;
From: ali@atilim.edu.tr; tag=18271
Call-ID: 3223842@new.hq.atilim.edu.tr
CSeq: INVITE 12921
Contact: <sip:jan@minibus.edu.tr>;
expires:7200
```

Ali's UA would then initiate a new INVITE-request (19) to the new SIP address in the Contact header field and follow the steps in Figure 1.

```
INVITE sip:jan@metu.edu.tr SIP/2.0
Via: SIP/2.0/UDP new.hq.atilim.edu.tr,rport
To: Jan<sip:jan@metu.edu.tr>
From: Ali <sip:ali@atilim.edu.tr>;
tag=18271
Call-ID: 3223842@new.hq.atilim.edu.tr
CSeq: 12921 INVITE
Contact: <sip:ali@new.hq.atilim.edu.tr>
```

Response will be the same as 9) but with the change in the destination addresses so we call it (9')

Ali finally terminates the session with the following message (20):

```
BYE sip:jan@mp1.south.metu.edu.tr SIP/2.0
Via: SIP/2.0/UDP new.hq.atilim.edu.tr
To: Jan<sip:jan@metu.edu.tr>; tag=129991
```

```
From: Ali <ali@atilim.edu.tr>; tag=18271

Call-ID: 3223842@new.hq.atilim.edu.tr

CSeq: 12923 BYE

Contact: <sip:ali@new.hq.atilim.edu.tr>
```

To finalize the session termination, Jan's terminal provides a response (21) to Ali's BYE request

```
SIP/2.0 200 Success

To: Jan<sip:Jan@metu.edu.tr>; tag=129991

From: Ali <ali@atilim.edu.tr>; tag=18271

Call-ID: 3223842@new.hq.atilim.edu.tr

CSeq: 12923 BYE
```

Although we have proposed this protocol we could not implement it because it requires a test bed which we did not have the infrastructure to implement. To the best of our knowledge there isn't a practical way to test the proposed protocol and we could not access any simulator to simulate this configuration.

CHAPTER 6

Conclusion

In this work we have looked at NAT traversal problem in VoIP applications. We proposed a protocol for NAT traversal using pure SIP messages in case of mobility during VoIP session.

Our solution is based on Boulton and Rosenberg [26] [Appendix B] work and we have extended this study to include Mobility . In our solution we make use of the fact that the mobile node already knows the location of the other party since it does not move. We have only considered the signaling part in this work.

As future work Mobile IP solutions for NAT traversal may be investigated and the two solutions can be compared.

APPENDIX A

RPORT Parameter

"rport" parameter allows a client to request that the server send the response back to the source IP address and port where the request came from. The "rport" parameter is analogous to the "received" parameter, except "rport" contains a port number, not the IP address.

A.1 Syntax

The syntax for the "rport" parameter is:

```
response-port = "rport" [EQUAL 1*DIGIT]
```

This extends the existing definition of the Via header field parameters, so that its BNF now looks like:

```
via-params =     via-ttl / via-maddr
                 / via-received / via-branch
                 / response-port / via-extension
```

A.2 Example

A client sends an INVITE to a proxy server which looks like, in part:

```
INVITE sip:user@example.com SIP/2.0
Via: SIP/2.0/UDP 10.1.1.1:4540;rport;branch=z9hG4bKkjshdyff
```

This INVITE is sent with a source port of 4540 and a source IP address of 10.1.1.1. The proxy is at 192.0.2.2 (proxy.example.com), listening on both port 5060 and 5070. The client sends the request to port 5060. The request passes through a NAT on the way to the proxy, so that the source IP address appears as 192.0.2.1 and the source port as 9988. The proxy forwards the request, but not before appending a value to the "rport" parameter in the proxied request:

```
INVITE sip:user@example.com SIP/2.0
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bKkjsh77
Via: SIP/2.0/UDP 10.1.1.1:4540;received=192.0.2.1;rport=9988;

branch=z9hG4bKkjshdyff
```

This request generates a response which arrives at the proxy:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy.example.com;branch=z9hG4bKkjsh77
Via: SIP/2.0/UDP 10.1.1.1:4540;received=192.0.2.1;rport=9988;
branch=z9hG4bKkjshdyff
```

The proxy strips its top Via header field value, and then examines the next one. It contains both a "received" parameter and an "rport" parameter. The server follows the rules specified in Section 4 and sends the response to IP address 192.0.2.1, port 9988, and sends it from port 5060 on 192.0.2.2:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.1:4540;received=192.0.2.1;rport=9988;
branch=z9hG4bKkjshdyff
```

This packet matches the binding created by the initial request. Therefore, the NAT rewrites the destination address of this packet back to 10.1.1.1, and the destination port back to 4540. It forwards this response to the client, which is listening for the response on that address and port. The client properly receives the response.

APPENDIX B

Best Current Practices for NAT Traversal for SIP

Traversal of the Session Initiation Protocol (SIP) and the sessions it establishes through Network Address Translators (NAT) is a complex problem. This appendix aims to provide an introduction to the paper that explains this problem [26].

NAT (Network Address Translators) traversal has long been identified as a large problem when considered in the context of the Session Initiation Protocol (SIP) and it's associated media such as Real Time Protocol (RTP). The problem is further confused by the variety of NATs that are available in the market place today and the large number of potential deployment scenarios.

The IETF has produced many specifications for the traversal of NAT, including STUN, ICE, rport, symmetric RTP, TURN, connection reuse, SDP attribute for RTCP, and others. These each represent a part of the solution, but none of them gives the overall context for how the NAT traversal problem is decomposed and solved through this collection of specifications. the mentioned document serves to meet that need.

The document attempts to provide a definitive set of 'Best Common Practices' to demonstrate the traversal of SIP and it's associated media through NAT devices. The

document does not propose any new functionality but does draw on existing solutions for both core SIP signaling and media traversal.

REFERENCES

[1] R. Arora, *Voice over ip: Protocols and standards*, http://www.cis.ohio-state.edu/ jain/cis788-99/voip_protocols/index.html.

[2] K. Nichols S. Blake F. Baker and D. Black, *Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers*, RFC 2474 (Dec. 1998).

[3] U. Black, *Voice over ip*, Prentice Hall PTR, Upper Saddle River, NJ, 2000.

[4] W. Almesberger J.Y. L. Boudec and T. Ferrari, *calable resource reservation for the internet*, in Proc. IEEE Conf. Protocols for Multimedia SystemsMultimedia Networking (Santiago, Chile), Nov. 1997.

[5] P.-F. Yang D. Shrader H. Sinnreich F. Mnard C. A. Polyzois, K. H. Purdy and H. Schulzrinne, *From pots to pans: A commentary on the evolution to internet telephony*, IEEE Internet Computing, vol. 3, no. 3 (May/June 1999).

[6] DataBeam Corp., *A primer on h.323 series standard*, URL: http://www.lotus.com/products/sametime.nsf/Menu/PDF/$FILE/h323 _primer-v2.pdf.

[7] MICOM Communications Corp., *Rsvp: Resource reservation protocol,*, http://www.micom.com/WhitePapers/rsvp/wprsvpte.htm.

[8] Nortel Corp., *Voice/fax over ip: Internet, intranet, and extranet*, http://www.alliancedatacom.com/voice-fax-over-ip.htm.

[9] I. Dalgic and H. Fang, *Comparison of h.323 and sip for ip telephony signaling*, in Proc. Photonics East (Sep. 20-22, 1999).

[10] B. Davie, *Ip qos for voice*, http://www.comsoc.org/confs/isit/99/Davie.pdf.

[11] J. Ryan Ed., *Voice over ip (voip)*, The Applied Technologies Group Inc., Natick, MA, 1998.

[12] The Internet Engineering Task Force, *Ietf home page*, http://www.ietf.org/.

[13] H. Schulzrinne S. Casner R. Frederick and V. Jacobsen, *Rtp: A transport protocol for real-time applications,*, RFC 1889 (Jan. 1996).

[14] BUR GOODE, *Voice over internet protocol (voip)*, IEEE Invited Paper.

[15] W. J. Goralski and M. C. Kolon, *Ip telephony*, McGraw-Hill, New York, NY, 2000.

[16] M. Handley and V. Jacobsen, *Sdp: Session description protocol*, RFC 2327 (Apr. 1998).

[17] R. Braden D. Estrin S. Berson S. Herzog and D. Zappala, *The design of the rsvp protocol*, ftp://ftp.isi.edu/pub/hpcc-papers/rsvp.final.report.ps.

[18] R. Braden Ed. L. Zhang S. Berson S. Herzog and S. Jamin, *Resource reservation protocol (rsvp), version 1: Functional specification*, RFC 2205 (Sep. 1997).

[19] C. Hitema, *Short term nat requirements for udp based peer-to-peer applications*, IETF Draft (Feb.2001).

[20] RealNetworks Inc., *Rtsp frequently asked questions,*, http://www.realnetworks.com/devzone/library/rtsp/faq.html.

[21] Symbol Technologies Inc., *Converging technologies for voice and data networks,*, ftp://sysmstore.longisland.com/Symstore/pdf/converge.pdf.

[22] M. Handley C. Perkins and E. Whelan, *Session announcement protocol*, nternet-Draft draft-ietf-mmusic-sap-v2-06.txt (Mar. 2000).

[23] Protocols.com, *H.323*, http://www.protocols.com/pbook/h323.htm.

[24] S. Zeadally F. Siddiqui N.DeepakMavatoor P. Randhawa, *Sip and mobile ip integration to support seamless mobility*, IEEE (2004).

[25] H. Schulzrinne A. Rao and R. Lanphier, *Real time streaming protocol (rtsp)*, RFC 2326 (1998).

[26] C. Boulton J. Rosenberg, *Best current practices for nat traversal for sip*, Internet-Draft (October 2004).

[27] J. Rosenberg and H. Schulzrinne, *Timer reconsideration for enhanced rtp scalability,*, in Proc. IEEE Infocom, San Francisco, CA (Mar. 29Apr. 1999).

[28] H. Schulzrinne E. Schooler and J. Rosenberg, *Sip: Session initiation protocol*, RFC 2543 (Mar. 1999).

[29] H. Schulzrinne, *Internet-media-on-demand: The real-time streaming protocol*, http://www.cs.columbia.edu/ hgs/teaching/ais/slides/RTSP1.pdf.

[30] _____, *Rtp profile for audio and video conferences with minimal control*, RFC 1890 (Jan. 1996).

[31] H. Schulzrinne and J. Rosenberg, *The ietf internet telephony architecture and protocols*, http://www.computer.org/internet/telephony/w3schrosen.htm.

[32] Hughes Software Systems, *Mgcp/megaco voip gateway solution*, http://www.hssworld.com/products/next_generation_networks /signallingvoipgateway/mgcpvoip/architecture.htm.

[33] _____, *Sip stack*, http://www.hssworld.com/products/protocolstacks/sip/sip_home.htm.

[34] A. S. Tanenbaum, *Computer networks*, 3 rd ed., Prentice Hall PTR, Upper Saddle River, NJ, 1996.

[35] Cisco Technologies, *Voice over ip for the cisco 3600 series software configuration guide*, Tech. report, http://www.combinet.com/univercd/cc/td/doc /product/software/ios113ed /113t/113t_1/voip/voipo ver.pdf.

[36] vicomsoft, *Network address translation faq*, White Paper, http://www.vicomsoft.com/knowledge/reference/nat.html.

[37] S. Blake D. Black M. Carlson E. Davies Z. Wang and W. Weiss, *An architecture for differentiated services*, RFC 2475 (Dec. 1998).

.