

Design of a practical Voice over Internet Protocol network for the Multi User Enterprise

**Jacob Bester Loubser
66548**

**A dissertation submitted in
fulfilment of the requirements for the**

**Magister Technologiae: Engineering: Electrical
Faculty of Engineering**

**Department Applied Electronics and Electronic
Communication**

Faculty of Engineering and Technology

**Vaal University of Technology
Vanderbijlpark
South Africa**

Supervisor: Dr. H.C.vZ. Pienaar

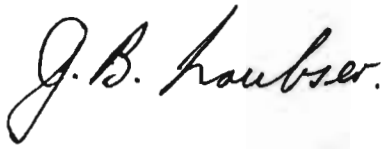
Date: June 2005

VAAL UNIVERSITY OF TECHNOLOGY	
Bib. No.	11005567
Item No.	11172290
Order No.	UNIPPA
2005-09-30	
Price:	R2000
Call No.	004.678 L01
LIBRARY STOCK	

**FOR
REFERENCE ONLY**

Declaration

I, Jacob Bester Loubser declare that this project is my own, unaided work. It is being submitted for the Magister Technologiae: Engineering: Electrical to the Department of Applied Electronics and Electronic Communication at the Vaal University of Technology, Vanderbijlpark. It has not been submitted before for any qualification or examination to any educational institution.

A handwritten signature in black ink, reading "J. B. Loubser." The signature is written in a cursive style with a large initial 'J'.

Jacob Bester Loubser

8 June 2005

Acknowledgements

- Dr. H.C. van Z. Pienaar for his leadership and encouragement with this project.
- Mr. W.J. van Z. Pienaar for his assistance with the layout of this document.

Abstract

This dissertation discusses the design and implementation of a voice over internet protocol system for the multi-user enterprise. It is limited to small to medium enterprises of which the Vaal University of Technology is an example. Voice communications over existing Internet protocol networks are governed by standards, and to develop such a system it is necessary to have a thorough understanding of these standards. Two such standards namely the International Telecommunications Unions H.323 and the Internet Engineering Task Force's SIP were evaluated and compared to each other in terms of their complexity, extensibility and scalability as well as the services they offer. Based on these criteria it was decided to implement a SIP system.

A SIP network consists of application software that act as clients and servers, as well as hardware components such as a proxy and redirect and registrar or location servers that allow users of this network to call each other on the data network. Gateways enable users of the network to call regular public switched telephone network numbers. A test network was set up in the laboratory that contained all the hardware and software components. This was done to understand the installation and configuration options of the different software components and to determine the suitability and interoperability of the software components. This network was then migrated to the network of the Vaal University of Technology which allowed selected users to test and use it. Bandwidth use is a major point of contention, and calculations and measurements showed that the codec being used during the voice call is the determining factor.

This SIP system is being used on a daily basis and the users report excellent audio quality between soft phones and soft phones, soft phones and normal telephones and even cellular phones.

Table of contents	Page
Declaration	ii
Acknowledgments	iii
Abstract	iv
List of figures	viii
List of tables	ix
List of abbreviations	x
Chapter 1 Introduction ✓	11
1.1 Background	11
1.2 Purpose of the study ✓	11
1.3 Problem statement ✓	11
1.4 Research methodology ✓	12
1.5 Importance of the research	12
1.6 Summary ✓	13
Chapter 2 Voice over Internet Protocol Standards	14
2.1 Introduction	14
2.2 The H.323 protocol	14
2.2.1 The H.323 protocol stack	14
2.2.2 The H.225.0 registration, admission and signalling protocol	15
2.2.3 Call signalling using H.225.0	17
2.2.4 The H.245 media and conference control protocol	17
2.2.5 Call setup with H.323	18
2.3 Session initiation protocol architecture	18
2.3.1 Session initiation protocol components	19
2.3.2 Session initiation protocol messages	20
2.3.3 Session initiation protocol operation	20
2.4 Session description protocol	21
2.5 Supporting protocols	23
2.6 Real time transport protocol	24

2.7	Real time control protocol	25
2.8	Real time streaming protocol	26
2.9	Resource reservation protocol	27
2.10	Summary	27

Chapter 3 Comparing H.323 and SIP for Internet Telephony **28**

3.1	Introduction	28
3.2	Complexity	28
3.2.1	H.323 protocol complexity	28
3.2.2	Session initiation protocol complexity	29
3.3	Extensibility	29
3.3.1	Extensibility of H.323	30
3.3.2	Session initiation protocol extensibility	31
3.4	Scalability	32
3.4.1	Scalability of H.323	32
3.4.2	Session initiation protocol scalability	33
3.5	Services	34
3.6	The choice between H.323 and session initiation protocol	35
3.7	Summary	36

Chapter 4 Voice over Internet Protocol Network Architecture and Design **37**

4.1	Introduction	37
4.2	Session initiation protocol network components	37
4.2.1	User agent	38
4.2.2	Proxy server	38
4.2.3	Redirect server	52
4.2.4	Registrar or location servers	54
4.3	Detailed session initiation protocol message flow	55
4.4	The network design and test environment	57
4.5	Design implementation	59
4.6	Convergence at the Vaal University of Technology network	60

4.7	Summary	62
Chapter 5 Measurements and results		63
5.1	Introduction	63
5.2	Review of relevant measurement theory	63
5.3	Voice over Internet protocol bandwidth calculation	64
5.4	Actual Voice over Internet protocol bandwidth measurements	68
5.5	Audio quality results	70
5.6	Summary	71
Chapter 6 Conclusion		72
6.1	Findings and implications	72
6.2	Reassessment of problem	73
6.3	Recommendations	74
6.4	Fields for future study	75
References		76
List of sources consulted		76
List of sources quoted		78
Annexure		80

List of figures

Figure 1: H.323 protocols stack	15
Figure 2: Internet telephony protocol stack	23
Figure 3: SIP network components	38
Figure 4: A stateful proxy model	40
Figure 5: Non-Invite Request-Response message flow	44
Figure 6: INVITE-non-2xx-ACK message flow	44
Figure 7: INVITE-200-ACK message flow	45
Figure 8: CANCEL processing	46
Figure 9: Message proxying without Record-Routing	47
Figure 10: Recursion on 3xx response	48
Figure 11: Parallel Forking	49
Figure 12: Sequential Forking	50
Figure 13: Authentication	51
Figure 14: Illegal loop	51
Figure 15: Spiral example	52
Figure 16: The request redirection process	54
Figure 17: SIP Registration Process	55
Figure 18: Detailed SIP message flow	56
Figure 19: The primary network design and test environment	57
Figure 20: Departmental or satellite campus network	59
Figure 21: Design implementation environment	60
Figure 23: Bandwidth occupied during SPX codec call	68
Figure 24: Bandwidth occupied during an iLBC codec call	69
Figure 25: Bandwidth occupied during a GSM codec call	69
Figure 26: Bandwidth occupied during a G.711a codec call	70
Figure 27: Bandwidth occupied during a G.711u codec call	70

List of tables

Table 1 : An example SDP description	22
Table 2: SIP and H.323 call control comparison	34
Table 3: Comparing H.323 with SIP	35
Table 4: SIP 3xx responses	53

List of abbreviations

A

ASN.1 – Abstract syntax notation 1
ARQ – Admission request message
ACF – Admission confirm message
ATM – Asynchronous transfer mode
API – Application programming interface

B

bps – Bits per second
BRQ – Bandwidth change request message
BGP – Border gateway protocol

C

Codec – Coder decoder
CPU – Central processing unit
CRC – Cyclic redundancy check

D

DNS – Domain name system

F

FXO – Foreign exchange office
FXS – Foreign exchange station

G

GRQ – Gatekeeper request message
GCF – Gatekeeper confirmation message
GRJ – Gatekeeper reject message

H

HTTP – Hyper text transfer protocol

I

IETF – Internet Engineering Task Force
ITU – International Telecommunications Union
ITU-T – International Telecommunications Union (Telecommunications sector)
IP – Internet protocol
IPX – Internetwork packet exchange
IANA – Internet Assigned Numbers Authority

K

kbps – Kilobits per second
kBps – Kilo bytes per second

L

LAN – Local area network

LCF – Location confirmation message
LRJ – Location reject message
LRQ – Location request message

M

MAC – Media access control
MC – Multipoint controller

P

PC – Personal computer
PSTN – Public switched telephone network
PPP – Point to point protocol
PEP – Protocol extensions protocol

Q

QoS – Quality of service

R

RAS – Registration admission and signalling
RRQ – Registration request
RFC – Request for comment
RSVP – Resource reservation protocol
RTP – Real time transport protocol
RTCP – Real time control protocol

S

SAP – Session announcement protocol
SDP – Session description protocol
SMTP – Simple mail transfer protocol
SIP – Session initiation protocol
SOHO – Small office home office
SSRC – Synchronization source indicator

T

TCP – Transmission control protocol
TSAP – Transport layer service access point
TTL – Time to live

U

UDP – User datagram protocol
URI – Uniform resource identifier
URL – Uniform resource locator
UAC – User agent client
UAS – User agent server

V

VoIP – Voice over Internet protocol
VUT – Vaal University of Technology

W

WAN – Wide area network

X Y Z

Chapter 1 Introduction

1.1 Background

Ever since Microsoft included a program called NetMeeting with its Windows Operating system it became possible for users to use voice communications over the Internet. This program provided simple personal computer (PC) to PC connections. This technology has evolved into an Internet Protocol (IP) telephony market which is capable of supporting not only voice but also video and data. New advanced communications applications have since been developed that allows PC-to-PC, PC to telephone and telephone to PC calls – all over an IP network. It is estimated that voice over Internet protocol (VoIP) has the potential to radically change the way communications are taking place at the moment, and it will have a profound impact on the telecommunications industry in the future. The International Data Corporation estimated that the total worldwide market for VoIP exceeded \$1.8 billion in the year 2003.

1.2 Purpose of the study

The purpose of this research is to demonstrate that with the current technology it will be possible to implement a voice communication system over the existing data network of the multi user enterprise. The Vaal University of Technology (VUT) data network is an example of such an enterprise network that could be used to demonstrate that voice communication over the data network is possible. Telecommunication companies have largely ignored the growth in data networks, and they continue to charge their costumers premium rates for services that are now for free with data networks. This work should prove that the consumer does not have to be reliant on premium rate services, and that with a little ingenuity an alternative is available.

1.3 Problem statement

For many years the basic assumption of telecommunication companies was that the customer must pay to use the scarce resources provided by them. However, in recent years the growth of the data network has far outstripped the growth of the conventional

voice network. If it is possible to communicate with a person on the other side of the world with the Internet why is it also not possible to communicate with that same person with voice over the Internet? VoIP is such a technology and it needs to be understood in order to be able to implement it properly. In order to understand voice communication over a data network, it is important to understand the standards that govern VoIP communications, as well as the hardware and software components that will ultimately make up the VoIP system.

1.4 The research methodology

The starting point of this work requires a thorough understanding of data networks and the rules governing this communication as well as the TCP/IP protocol suite that governs data communication on the Internet. The approach followed to arrive at a VoIP was as follows:

- An understanding of the standards that govern voice over data network communication in order to make informed decisions about the resources that is available for the implementation of such a system.
- The evaluation of the proposed system in terms of its complexity, extensibility and scalability as well as services offered in order to find the most suitable and cost effective solution.
- The purchase or design of the equipment for the demonstration of the technology.
- The design and testing of a voice over data network in the laboratory.
- Implementation and testing of the designed system at VUT.
- Evaluation of test results.
- Formulating recommendations.

1.5 Importance of the research

The following points highlight the importance of the research:

- It will be shown that the users of a data network of an enterprise, particularly the users of the data network at VUT, could use this network for communication.

- This technology may have great impact on future developments within the telecommunications industry.
- It will be shown to telecommunications companies that a paradigm shift is necessary if they want to stay competitive and profitable in the face of the rapid change of the needs of their costumers.
- It will also show that this is a new market niche that should be explored by telecommunications companies if they want to stay competitive in the market.
- The technology will decrease the cost of international telephone calls by up to 60 percent, for both the enterprise and the customer.

1.6 Summary

This chapter briefly provided background information on VoIP. The purpose of the study and current problems where indicated, and the methodology that will be used to conduct the research was stated. The importance of this research to the multi user enterprise and the telecommunication industry was also highlighted.

The next chapter addresses the standards governing VoIP systems so as to provide a thorough understanding of the operation of various VoIP implementations and systems.

Chapter 2 Voice over Internet Protocol Standards

2.1 Introduction

The International Telecommunications Union (ITU), with H.323 protocol as its recommendation, and the Internet Engineering Taskforce (IETF), with session initiation protocol (SIP) as its implementation, both decided on a layered approach for the design of the respective proposed voice over Internet protocol stack designs. This approach has, as its main aim, the simplification of a complex process, by breaking it up into distinct steps to simplify the entire process. These implementations have one common goal and that is that VoIP products from different vendors can interoperate. It is important to understand the operation of the various protocols since it will aid in making important decisions in solving the problem. The following protocols are discussed in order to be able to decide on the most suitable implementation.

- H.323
- SIP
- SDP
- RTP
- RTCP
- RTSP
- RSVP

2.2 The H.323 protocol

2.2.1 The H.323 protocol stack

H.323 is an ITU-T recommendation umbrella set of standards that defines the components, protocols, and procedures necessary to provide multimedia (audio, video, and data) communications over IP-based networks. Today it is the most mature and widely used standard and enjoys industry-wide support. In addition to control and call setup standards, H.323 also includes protocols for audio, video and data transmission (Arora 1999:3).

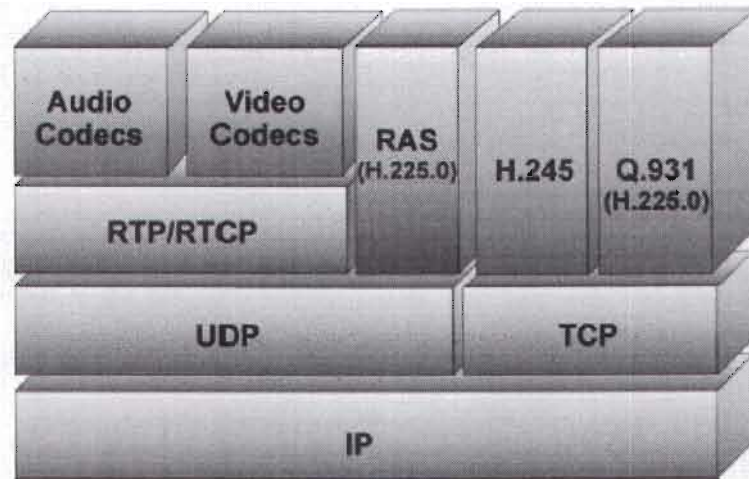


Figure 1: H.323 protocols stack (RADCOM 2003:7).

As shown by figure 1, the audio, video and registration, admission and signalling use the unreliable user datagram protocol (UDP). The data and control applications use the reliable transport control protocol (TCP) as the transport mechanism. H.323 provides three control protocols namely: Q.931/ H.225.0 call signalling, H.225.0 registration, admission and signalling (RAS), and H.245 media control (Arora 1999:5). Q.931/ H.225.0 are used with H.323 to provide signalling for call control while H.225.0 RAS channels are used to establish a call between two parties. After the call is established H.245 is used to negotiate the media streams (Arora 1999:6).

2.2.2 The H.225.0 registration, admission and signalling protocol

The RAS channel provides communication between endpoints and the gatekeeper. The RAS messages are sent using UDP which is inherently unreliable and therefore, provisions is made for timeouts and retry counts for messages (Arora 1999:6). The procedures that are defined by the RAS channel are:

- The gatekeeper discovery process is used by the endpoint to discover the gatekeeper with which it should register. The discovery process involves the broadcasting of a gatekeeper request (GRQ) message by the endpoint. At this time, one or more gatekeepers may respond with a gatekeeper confirmation (GCF) message indicating the willingness of the gatekeeper to act as the gatekeeper for that endpoint. Included in the response is the transport address of the gatekeeper's RAS channel. Should a gatekeeper reject the endpoints request, it sends a gatekeeper reject (GRJ) message. If

more than one gatekeeper responds with a gatekeeper confirmation (GCF) message it is up to the endpoint to choose the gatekeeper with which it will register. If no responses are received within the timeout interval, the endpoint can rebroadcast GRQ message.

- Upon endpoint registration, the endpoint joins a zone, and sends its transport and alias addresses to the gatekeeper. All endpoints usually register with the gatekeeper that was identified during the discovery process. The endpoint must send a registration request (RRQ) to the gatekeeper RAS channel transport address. The endpoint is already aware of the gatekeeper's address that was obtained during the discovery process, and uses the RAS channel TSAP identifier.
- If an endpoint or gatekeeper has an alias address for an endpoint, and its contact information is unknown, a location request message (LRQ) is issued. The gatekeeper with whom the endpoint is registered responds with a location confirmation message (LCF). This message contains the contact information of the endpoint location or the address of the gatekeeper with which the endpoint is registered. All gatekeepers with whom the endpoint is not registered will issue a location reject message (LRJ) once they receive a location request message (LRQ) on the RAS channel.
- Admissions, bandwidth change, status and disengagement are transmitted using the RAS channels. These messages are transmitted between endpoints and gatekeepers and are used for the management of admission control and bandwidth. The control signals that are being used here are: Firstly, an admission request message (ARQ) which specifies the requested call bandwidth. Secondly, the admission confirm message (ACF) used by the gatekeeper to reduce the requested bandwidth. Thirdly, the bandwidth change request message (BRQ) that may be used by either endpoints or gatekeepers to attempt to modify the call bandwidth during a call.

Other RAS channel tasks include:

- Endpoint location.
- Admission, bandwidth change, status and disengagement.

- Access tokens can provide privacy by shielding an endpoint's transport address and alias address information from the calling party and ensure that calls are routed properly through H.323 entities (Väänänen 1999:26).

2.2.3 Call signalling using H.225.0

The call signalling channel is used to carry H.225.0 control messages. If a network does not have any gatekeepers, call signalling messages are passed directly between calling and called endpoints using the call signalling transport address. At this point it must be assumed that the calling endpoint knows the call signalling transport address of the called endpoint and that direct communication is possible. In networks where a gatekeeper is present, the initial admission message exchange takes place between the calling endpoint and the gatekeeper using the gatekeeper's RAS channel transport address (Arora 1999:7). This signalling exchange uses the reliable TCP protocol.

- Channel signalling and routing messages may be exchanged in two ways. The first way is gatekeeper routed call signalling where the call signalling messages are routed via the gatekeeper from the endpoints. The second way is direct endpoint call signalling where call signalling messages are exchanged directly between endpoints. Admission messages are exchanged with the gatekeeper using the RAS channel, followed by an exchange of call signalling messages using a call signalling channel. This is followed with the establishment of the H.245 control channel (Arora 1999:7).
- Control channel routing can be routed using one of two methods. The first method establishes the H.245 control channel directly between endpoints. The second method establishes the H.245 control channel via the gatekeeper (Arora 1999:7).

2.2.4 The H.245 media and conference control protocol

H.245 media control protocol is used after the call has been established. H.245 is used to negotiate and establish all of the media channels carried by RTP/RTCP (Arora 1999:7). H.245 provides the following functionality:

- Appointing master and slave: H.245 appoints a multipoint controller (MC) which is responsible for central control in the event of a call being extended to a conference.

- Capability exchange: H.245 is used to negotiate the capabilities after a call has been established. This exchange can take place any time during a call, which allows for renegotiation at any time.
- Media channel control: After conference endpoints have exchanged capabilities logical channels of media may be opened or closed by them. H.245 describes media channels as logical channels.
- Conference control: H.245 provides conference endpoints with mutual awareness and describes the media flow model between all the endpoints.

2.2.5 Call setup with H.323

Arora (1999:7) states that the process of setting up a call with H.323 can be broken up into a number of distinct steps:

- Discover a gatekeeper that is able to manage the endpoint.
- Register the endpoint with the gatekeeper.
- Endpoint goes into call setup phase.
- Capability exchange between endpoint and gatekeeper takes place.
- Call is established.
- When a call is complete the endpoint terminates the call. This termination can also be initiated by the gatekeeper.

2.3 Session initiation protocol architecture

Session initiation protocol (SIP) is a typical client server protocol similar in syntax and semantics to hyper text transfer protocol (HTTP), where requests are generated by the client and then sent to the server. The server, in turn, processes the request and returns a response to the client. The sum total of the request and the response is known as a transaction. SIP has INVITE and ACK messages which define the process of opening a reliable channel over which call control messages may be passed. SIP makes minimal assumptions about the underlying transport protocol. It can directly use any datagram or stream protocol, with the only restriction that a whole SIP request or response has to be either delivered in full or not at all. SIP can thus be used with UDP or TCP in the Internet, and with X.25, ATM, CLNP, TP4, IPX or PPP elsewhere. This protocol itself

provides reliability and does not depend on TCP for the provision of reliable transport services. SIP depends on the session description protocol (SDP) for carrying out the negotiation for codec identification. SIP supports session descriptions that allow participants to agree on a set of compatible media types. It also supports user mobility by substituting and redirecting requests to the user's current location (Arora 1999:8). The services that SIP provides include:

- User Location: Determination of the end system to be used for communication.
- Call Setup: Ringing and establishing call parameters at both called and calling party.
- User Availability: Determination of the willingness of the called party to engage in communications.
- User Capabilities: Determination of the media and media parameters to be used.
- Call handling: Transfer and termination of calls.

2.3.1 Session initiation protocol components

According to Arora (1999:8) a SIP system consists of two components namely, user agents and network servers:

- A user agent is an end system that acts on behalf of the user. There are two parts to the user agent: the client and the server. The client portion is known as the user agent client (UAC) and the server portion as the user agent server (UAS).
- Network Servers, can be classified into three categories. Firstly, a registration server that contains and receives updates of the current location of users. Secondly, a proxy server that, upon receiving requests, redirects the requests to the next-hop server which has more information about the location of the called party. The next-hop server might be another proxy server, a user agent server, or a redirect server (Schulzrinne 1999:3). Thirdly, a redirect server that, upon receiving requests, determines the next-hop server and returns the address of the next-hop server to the client instead of forwarding the request. The primary function of proxy and redirect servers is call routing which is the determination of the set of servers that must be traversed to complete the call. A proxy or redirect server can use any means available to determine the next-hop server, including the executing of programs and consultation of databases. A SIP proxy server can also fork requests, by sending

copies to multiple next-hop servers at once. This enables a call setup request to try many different locations at once. The first location to answer will be connected with the calling party (Schulzrinne 1999:3).

2.3.2 Session initiation protocol messages

As with HTTP, the client requests invoke commands on the server (Schulzrinne 1999:3). A number of messages are defined by SIP and are being used for communication between the client and the SIP server, these messages are:

- INVITE for inviting a user to a call.
- BYE for terminating a connection between the two end points.
- ACK for reliable exchange of invitation messages.
- OPTIONS for getting information about the capabilities of a call.
- REGISTER gives information about the location of a user to the SIP registration server.
- CANCEL for terminating the search for a user.

2.3.3 Session initiation protocol operation

All parties are identified by SIP addresses. When first making a SIP call the caller needs to locate the appropriate server, and having done so, send it a request. The called party can either be reached directly or indirectly through the redirect server (Arora 1999:9). The process of locating a user and setting up a call using SIP is:

- SIP addressing can be used to identify hosts. It is often in the form of a SIP universal resource locator (URL) for example sip:username@host. This SIP address can designate an individual or a group.
- A client can locate a SIP server by sending a request to a SIP proxy server or it can send it directly to the IP address and port corresponding to the universal resource identifier (URI).
- A SIP transaction consists of a request and the response triggered by that request. Once the host part of the request URI has been resolved to a SIP server, the client sends a request to that server. The transport mechanism used can be either the reliable transmission control protocol (TCP) or the unreliable user datagram protocol (UDP).

- A SIP invitation consists of two requests namely an INVITE followed by an ACK. The purpose of the INVITE request is to ask the called party to join a particular conference or to join a two way conversation. After the called party has agreed to participate, the caller confirms that decision by sending an ACK request. The INVITE request contains a session description containing enough information for the called party to join the session. If the called party wishes to join the session it responds to the invitation by sending a similar session description.
- A called party may be very mobile, and the new locations of the called party can be dynamically registered with the SIP server. When the SIP server is queried about the location of the called party the server returns a list of possible locations. This list will be generated by a location server which will pass it on to the SIP server.
- Changing the parameters of an existing session can be accomplished by reissuing the INVITE message using the same call ID but with a new body to convey the new information.

2.4 Session description protocol

Although strictly speaking session description protocol (SDP) is a supporting protocol to SIP, its function is so closely related that it can be viewed in conjunction with SIP. The function of SDP is to describe multimedia sessions, for both telephony and for distributed applications like “Internet radio” (Schulzrinne 1999:4). Information included in the protocol is:

- Media streams.

A multimedia session can contain many media streams, which may include a number of audio streams, and a video stream. This information is conveyed by SDP in the format the number and type of each media stream. Currently audio, video, data control and application steam types are defined.

- Addresses.

For each stream, destination addresses (unicast or multicast) must be indicated. The addresses for different media streams may differ in order for a user to receive audio on a low-delay Internet telephone application.

- Ports.

Here the UDP port numbers are indicated for sending and receiving of the different streams.

- Payload types.

The media formats supported during a particular session are conveyed.

- Start and Stop times.

For broadcast type sessions like a television program, the start, stop, and repeat times of the session are conveyed.

- Originator.

For broadcast type sessions, the session description includes the originator of the session as well as the contact information in case of technical difficulties.

All this information is conveyed with SDP in a simple text based format. When a call is setup using SIP, the INVITE message contains a SDP body which describes the session parameters acceptable to the caller. The response from the called party contains a modified version of this description, which includes the capabilities of the called party (Schulzrinne 1999:5).

Table 1 : An example SDP description (Schulzrinne 1999:5).

v=0
o=g.bell 87728 8772 IN IP4 132.151.1.19
s=Come here, Watson!
u=HTTP://www.ietf.org
e=g.bell@bell-telephone.com
c=IN IP4 132.151.1.19
b=CT:64
t=3086272736 0
k=clear:manhole cover
m=audio 3456 RTP/AVP 96
a=rtpmap:96 VDVI/8000/1
m=video 3458 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait

Table 1, is an example of a typical session description. The **v** line is used as a version identifier for the session. The **u** and **e** lines indicate the universal resource locator (URL), and e-mail address for more information about the session. The **c** line indicates the address for the session, the **b** line the bandwidth which is 64kbits/sec in this case and the **t** line the start and stop times. If this value equals to 0 it means that the session continues indefinitely. The **k** line contains the encryption key for the session. The three **m** lines identify a particular media stream, the port number for that stream, the protocol and a list of payload types. The **a** line specifies an attribute (Schulzrinne 1999:5).

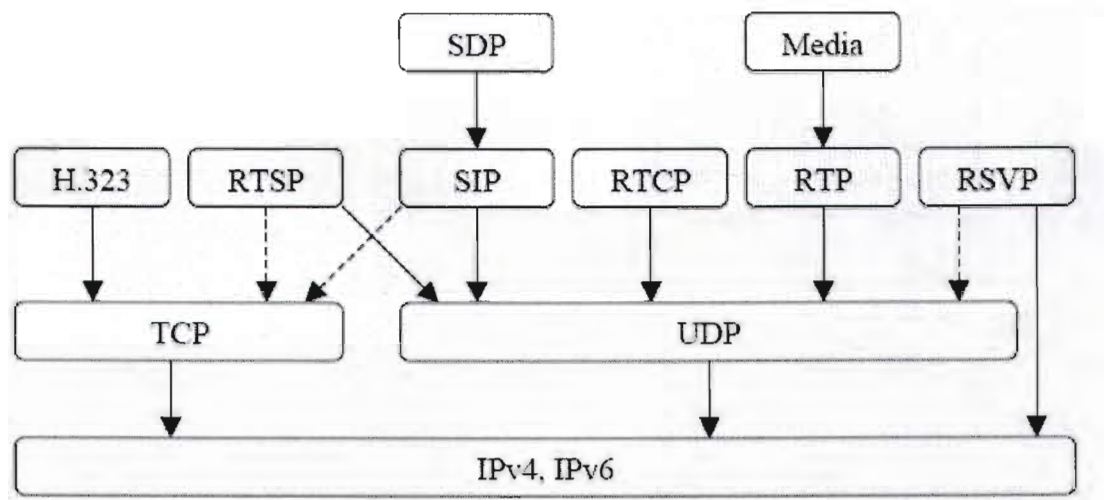


Figure 2: Internet telephony protocol stack. (Fingal 1999:10).

2.5 Supporting protocols

Both SIP and H.323 work in conjunction with real time transport protocol (RTP) and real time control protocol (RTCP). Implementing a number of different protocols, each serving a particular function, allows for modularity, flexibility, simplicity, and extensibility. End systems or network servers that only provide a specific service need only implement that particular protocol, without interoperability problems. Furthermore, protocol components can be reused in other applications, avoiding re-invention of specific functions in each application. The protocols shown and the rich services they offer are just a part of the picture that makes up Internet telephony and are all aimed at providing deterministic delivery in an inherently non-deterministic network environment

(Schulzrinne 1998:2). Figure 2 shows the relationship of these protocols to the signalling protocols discussed earlier in the chapter.

2.6 Real time transport protocol

As the name implies, real time transport protocol (RTP) supports the transport of real-time media over packet networks. The transport process involves the process of taking the bit stream generated by the media encoder, breaking it into packets, sending the packets over the network, and then recovering the bit stream at the receiver. This is a complex process since packets can be lost, delayed (by variable amounts) and moved out of order in the network. The protocol must allow the receiver to detect such losses. Timing information must also be transmitted so that the receiver can correctly compensate for variability in delay or jitter (Schulzrinne 1999:6).

To provide assistance in this function, RTP defines the formatting of the packets transmitted. The packets contain the RTP payload, the media information, and a RTP header. This header contains the necessary information that will allow the receiver to reconstruct the media. RTP also specifies how the codec bit streams are broken up into packets. It is important to note that RTP does not reserve resources in the network to avoid the loss of packets and jitter; in contrast it allows the receiver to recover in the presence of packet loss and jitter. RTP is effectively the “heart” of Internet telephony applications which is responsible for moving the actual voice between participants. The signalling protocols are used to establish the parameters for RTP transport (Schulzrinne 1999:6).

Specific functions provided by RTP are:

- Sequencing: Each RTP packet contains a sequence number, which is used for loss detection and compensating for re-ordering.
- Intra-media synchronization: Packets within the same stream may suffer different delays. Play-out buffers are used by applications to compensate for delay jitter, and these are measured by timestamps provided by RTP.
- Payload identification: In the Internet network, conditions such as packet loss and delay vary constantly. Speech and video codec's vary in their ability to work properly

under these conditions. It is therefore desirable to dynamically change the encoding for the media as conditions change. In order to achieve this, RTP contains a payload type identifier in each packet that describes the encoding of the media.

- Frame indication: Video and audio are sent in units called frames, and it is necessary to indicate to the receiver the beginning or end of a frame, in order to aid in synchronized delivery to higher layers. This is accomplished with a frame marker bit.
- Source identification: During a multicast session where many users are participating there must be a way of indicating which participant sent which packet. The synchronization source indicator (SSRC) is used for this purpose.

2.7 Real time control protocol

RTP has a companion control protocol, called real time control protocol (RTCP) which provides additional information to session participants (Schulzrinne 1999:6). The additional information provided is:

- Quality of service (QoS) feedback: RTCP is used by receivers in a session to report back on the quality of the reception from each sender. Information included is the number of lost packets, jitter, and round trip delays. This information is used by senders for applications which adjust encoding rates and other parameters based on this feedback information.
- Inter-media synchronization: Video and audio are often carried in separate packet streams, due to the need for flexibility. However they need to be synchronized at the receiver to provide “lip sync”. RTCP provides the necessary information required for the synchronization of sources, even when originating from different servers.
- Identification: RTCP packets contain information such as the e-mail address and telephone number of participants. This allows participants to learn the identities of other participants in the session.
- Session control: RTCP allows participants to indicate that they are leaving a session by using a BYE RTCP packet.

2.8 Real time streaming protocol

Real time streaming protocol (RTSP) is used to control a stored media server. A stored media server is capable of playing pre-recorded media from its disk to the network, and recording multimedia content to its disk. A client can instruct the server to play, record, fast-forward, and rewind and pause a message. The client can also configure the server with the IP addresses, UDP ports and speech codecs to use to deliver the media. The media is sent from the server using RTP (Schulzrinne 1999:7).

The applications of stored media in Internet telephony are:

- The content of a conference can be recorded for future reference.
- Media can be played into an existing conference.
- Media servers can be used to record voicemail. RTSP clients can use the protocol to playback the message. RTSP can also be used to record incoming and outgoing voicemail messages.

Schulzrinne (1999:7) lists the steps that a client needs to execute to get playback content from a media server as follows:

- Obtain the presentation description that counts the various components of the session. For each component the description defines the media parameters needed to decode the component, including the codec type and frame rate. The media description may be obtained by issuing a DESCRIBE request to the server, or through a web page.
- Setup the server: Once a description has been obtained the client issues a SETUP request to the server. By doing this, the destination where media should be delivered is determined. The destination includes the IP address, port numbers, protocols, time to live (TTL), and number of multicast layers. In response to the SETUP message the server provides a session ID which is used in further requests.
- Issue media requests: After the setup, the client can issue media requests to the server, which may include operations such as PLAY, RECORD, and PAUSE.
- Teardown: Once the server interaction is complete the client issues a TEARDOWN request, which destroys all information about the session.

2.9 Resource reservation protocol

The most important factors that hinder the voice-data convergence are network delay and QoS. Resource reservation protocol (RSVP) is the solution that enables a packet-switched network to emulate a deterministic circuit-switched voice network. With RSVP, voice communication with a tolerable delay is possible on data networks. RSVP requests resources in one direction only, therefore senders and receivers are treated as logically distinct entities, although an application process may act as sender and receiver at the same time. RSVP is not a routable protocol but it has been designed to operate with unicast and multicast routing protocols. RSVP places the responsibility on the receiver for requesting a specific QoS. A QoS request from a receiver host application is passed to the local RSVP process. RSVP then carries this request to all nodes on the reverse data path to the data source (Arora 1999:15).

RSVP attributes include:

- Receiver oriented.
- Unicast and multicast supported.
- Maintains soft state routers in hosts, which allows for dynamic membership changes.
- Transparent operation through routers that do not support it.

2.10 Summary

This chapter addressed the various protocols governing VoIP communications in detail. This was done in order to get a thorough understanding of the operation of the protocols so that important insight could be obtained about the operation of a particular protocol. The underlying support protocols were also addressed in order to get a good understanding of the delivery and session mechanisms which make an inherently nondeterministic data network more deterministic for voice communication.

The next chapter compares the two main VoIP protocols in terms of predefined criteria so that the most suitable solution to the problem can be chosen.

Chapter 3 Comparing H.323 and Session Initiation Protocol for Internet Telephony

3.1 Introduction

Two VoIP standards have been discussed up to this point. The two major standards are the ITU recommendation H.323 and the IETF SIP. These two protocols have one common goal, but attack the problem from two very different angles. In order to provide useful services, Internet telephony requires a set of control protocols for the establishment of a connection, the exchange of capabilities and conference control. In this chapter, these two protocols will be compared in terms of complexity, extensibility, scalability, and services in order to find the best solution to the problem.

3.2 Complexity

3.2.1 H.323 protocol complexity

The ITU H.323 recommendation defines protocols and procedures for multimedia communications on the Internet. Included is H.245 for control, H.225.0 for connection establishment, H.323 for large conferences, H.450.1/2/3 for supplementary services, H.235 for security and H.246 for interoperability with circuit switched services. H.323 leans heavily on the ITU multimedia protocols that preceded it. The encoding mechanisms, protocol fields and basic operation are simplified versions of Q.931 ISDN signalling protocol. H.323 is quite a complex protocol with the sum total of the base specification being 736 pages. H.323 uses a binary representation of its messages based on abstract syntax notation 1 (ASN.1). Generally ASN.1 requires special code-generators to parse. This complexity also is due to the fact that several protocol components are being used. There is no clean separation of the components, and many services require interactions between several components. Firewall traversal is also complicated because firewalls must act as application level proxies, parsing the entire message to arrive at the required fields. H.323 also provides for an array of options and methods to accomplish a single task. The specification had to be compatible with previous versions, and supported by firewalls, end systems, gatekeepers and gateways (Schulzrinne 1998:1).

Another aspect that makes H.323 complex is the duplication that exists. In particular, H.323 uses RTP and RTCP to provide feedback and conference control functions. H.245 also provides its own mechanisms for feedback and conference control functions. Thus the H.245 mechanism is redundant (Schulzrinne 1998:2).

3.2.2 Session initiation protocol complexity

SIP developed by the IETF, takes a different approach to Internet telephony signalling by “borrowing” from HTTP. SIP reuses many of the header fields, encoding rules, error codes, and authentication mechanism of HTTP (Schulzrinne 1998:1).

The entire SIP specification with its call control extensions and session description protocols totals only 128 pages. SIP has only 37 headers each with small number of values and parameters, but which contain more information. A basic but interoperable SIP Internet telephony implementation can get by with only four headers namely To, From, Call-ID, and CSeq and only three request types namely INVITE, ACK, BYE (Schulzrinne 1998:1).

SIP messages are encoded as text messages much like HTTP, and RTSP. This leads to simple parsing and generation particularly if a powerful text processing language such as Perl is being used. The text based encoding also aids debugging, which allows for manual editing and perusing of messages (Schulzrinne 1998:1).

Firewall traversal with SIP is easy since a single request is used, that contains all the necessary information to accomplish the task of traversal (Schulzrinne 1998:1).

3.3 Extensibility

This is the key metric for measuring an IP telephony protocol. As with any heavily used service, the features available evolve with time as new applications and technologies are developed. Hence compatibility between versions becomes a key factor. Because the Internet is an open, distributed and an ever evolving entity, it is to be expected that extensions to IP telephony protocols will be widespread and uncoordinated. It is,

therefore, critical to build in powerful extension mechanisms from the outset (Schulzrinne 1998:2).

3.3.1 Extensibility of H.323

Extensibility mechanisms provided with H.323 are generally nonstandardParam fields that are placed in various locations in the ASN.1. These fields contain a vendor code, followed by an opaque value which has a meaning only for that specific vendor. This allows vendors to develop their own extensions, but it also has some limitations. In the first place, extensions are limited only to those places where a non-standard parameter has been added. In the second instance, H.323 has no mechanism for allowing terminals to exchange extension support information. Since the values in non-standard parameters are not self describing, it limits interoperability among terminals from different manufacturers (Schulzrinne 1998:2).

To add to this, H.323 requires full backward compatibility between versions, which will cause the size of the encodings to increase as various features come and go (Schulzrinne 1998:2).

A critical issue for extensibility is audio and video codecs. With H.323 each codec must be centrally registered and standardized. At this time only ITU developed codecs have code points and many of these carry significant intellectual property. This presents a significant barrier to small players in the market in terms of cost (Schulzrinne 1998:2).

Another aspect that needs to be considered is modularity. Internet telephony demands a large number of different functions such as basic signalling, conference control, quality of service and so on. It must be assumed that mechanisms for accomplishing these functions will evolve over time. It is therefore critical that these functions be apportioned to separate modular components. H.323 is not very modular, since it defines a vertically integrated protocol suite for a single application. The mix of services provided is intertwined within the various sub-protocols within H.323 (Schulzrinne 1998:2).

3.3.2 Session initiation protocol extensibility

SIP has built in a rich set of extensibility and compatibility functions, after learning from well established protocols such as HTTP and simple mail transfer protocol (SMTP). Unknown headers and values are ignored by default. When a server receives a request it checks the list of named features in the Requires header. If any of it is not supported the server returns an error code, with a list of features it does understand. The client can then adapt to a simpler operation. New features can be registered with the Internet Assigned Numbers Authority (IANA), which means that any developer can create new features in SIP, and then simply register names for them. Compatibility is maintained across different versions (Schulzrinne 1998:2).

Since SIP is similar to HTTP, mechanisms that were developed for HTTP extensibility can also be used for SIP. One of these is the protocol extensions protocol (PEP), which contains pointers to the documentation for various features within the HTTP messages themselves (Schulzrinne 1998:2).

A critical issue for extensibility is audio and video codecs. SIP uses the session description protocol (SDP) to transmit the codecs supported by a terminal in a session. Codecs are identified by string names, and this can be registered by any person or organization with IANA. This has the result that SIP can work with any codec, and the name of the codec, and contact information can be determined by other implementations by contacting IANA (Schulzrinne 1998:2).

SIP allows new services to be defined by using a few powerful third-party call control mechanisms. These mechanisms allow a third party to instruct another entity to create and destroy calls to other entities. These mechanisms can be used to supply a variety of services such as blind transfer, operator assisted transfer, and various forwarding variations (Schulzrinne 1998:2).

SIP is reasonably modular, since it encompasses basic call signalling, user location, and registration. QoS, directory access, service directory, session content description are all

orthogonal and can be found in separate protocols. Due to SIP's modularity it can be used in conjunction with H.323. A user can locate another user using SIP's multi-hop search facilities, and once the user is located the response can be redirected to an H.323 URL (Schulzrinne 1998:3).

3.4 Scalability

Scalability can be defined on a number of different levels. It refers to the capability of the protocol to "grow" with the network, in terms of a number of factors (Schulzrinne 1998:3).

These factors are:

- Large number of Domains.
- Server processing.
- Conference sizes.
- Feedback.

3.4.1 Scalability of H.323

Large number of Domains:

Originally H.323 was planned to work on a single LAN, and therefore factors such as wide area addressing and user location were not important. Therefore, for large numbers of domains and complex location operations, H.323 has scalability problems, since it has no easy mechanism to perform loop detection in complex, multi-domain searches (Schulzrinne 1998:3).

Server Processing:

An H.323 system requires that both telephony gateways and gatekeepers handle calls from a large number of users. H.323 requires gatekeepers to be stateful. The call state must be kept for the entire duration of the call. To add to this, the connections used are TCP based, which results in a gatekeeper holding its TCP connection for the duration of the call (Schulzrinne 1998:3). Furthermore, a gatekeeper or gateway needs to process the signalling message for each call. Therefore, the simpler the signalling the more calls can be supported, which is not the case with H.323 (Schulzrinne 1998:3).

Conference sizes:

A multi-party conference with multicast data distribution is supported by H.323. In order to do this, a multipoint controller (MC) is required for the processing of all the signalling for even the smallest conference. This causes several difficulties, firstly if the user providing the MC functionality leaves the conference and terminates his/her application, the entire conference terminates. Secondly since the MC and gatekeeper functionality is optional, H.323 cannot even support three party conferences in some cases. To support even larger conferences H.323 protocol defines additional procedures. The result is that three distinct mechanisms exist to support conferences of different sizes (Schulzrinne 1998:3).

Feedback:

Procedures in H.245 define how receivers control media encodings, transmission rates, and error recovery. This method of feedback makes sense in point-to-point scenarios, but is not functional in multi-point conferencing (Schulzrinne 1998:4).

3.4.2 Session initiation protocol scalability

Large number of Domains:

SIP makes use of a loop detection algorithm similar to the one used in border gateway protocol (BGP), which allows it to be done in a stateless manner (Schulzrinne 1998:3).

Server Processing:

It is required from SIP servers and gateways to handle many calls. A transaction through several servers and gateways can be either stateful or stateless. With the stateless model, a server receives a call request, performs some operation, forwards the request and forgets about it. There is enough state information contained within SIP messages to allow for the response to be forwarded correctly. To add to this SIP can be carried on either TCP or UDP. If UDP is used, no connection state is required. The simpler the signalling the faster it can be processed. Since SIP is a simpler process it should be able to handle more calls per second compared to H.323 on a particular computer (Schulzrinne 1998:3).

Conference sizes:

SIP scales well to all different conference sizes, since there is no requirement for a central MC, resulting in conference coordination that is fully distributed. This results in an improvement in scalability and complexity. It can use UDP as well as TCP. Native multicast signalling is supported by SIP allowing a single protocol to scale from sessions with two to millions of members (Schulzrinne 1998:4).

Feedback:

SIP relies on RTCP for providing feedback reception quality. Like SIP, RTCP operates in a fully distributed fashion. The feedback provided, scales automatically from a two-user point-to-point conference to a broadcast-type conference with millions of participants (Schulzrinne 1998:4).

3.5 Services

Table 2 lists some of the call control services supported by both SIP and H.323 (Schulzrinne 1998:4).

Table 2: SIP and H.323 call control comparison (Schulzrinne 1998:4).

Feature	SIP	H.323
Blind transfer	Yes	Yes
Operator assisted transfer	Yes	No
Hold	Yes using SDP	Not yet
Multicast conferences	Yes	Yes
Multi-unicast conferences	Yes	Yes
Bridged conferences	Yes	Yes
Forward	Yes	Yes
Call park	Yes	No
Direct call pickup	Yes	No

A direct comparison is a little difficult as new services are always being added to both, and it is possible that more services will be added as time passes (Schulzrinne 1998:4). In addition to call control services, capabilities exchange services are provide by both SIP

when used with SDP and H.323. However H.323 provides a much more comprehensive set of functionality. SIP on the other hand provides more comprehensive support for personal mobility services. When a caller contacts a called party the called party can redirect the caller to a number of locations which can be arbitrary URLs. SIP also allows for multi-hop searches of a user. If the user is not currently residing at a particular SIP server this server will then proxy the request to one or more additional servers, until the party is contacted. The support provided by H.323 for this kind of mobility is much more limited. It cannot be used to express preferences nor can the caller express preferences in the original call invitation. H.323 does not allow for gatekeepers to proxy requests to multiple servers. H.323 supports a number of conference control services, as opposed to SIP that does not provide this control, but rather relies on other protocols for this service. Simple conference control functions are provided by RTCP (Schulzrinne 1998:4).

3.6 The choice between H.323 and session initiation protocol

Table 3 provides a comparison of the main features of H.323 and SIP which formed the basis for the decision to use a particular implementation.

Table 3: Comparing H.323 with SIP.

Criteria	H.323	SIP
Complexity	Very complex	Simple
Extensibility	Very difficult	Easy
Scalability	Difficult	Easy
Services	Fair	Multiple

Based on this analysis it was decided to implement a VoIP system based on SIP. Given the availability of resources, the relative simplicity of operation and the potential to scale well, SIP is well suited for experiments, on a small network, and it also has the potential to grow into networks of enterprise proportions. During the experimental phase, the nature of the messages will make it easy to do fault finding. During the operational phase, the simplicity of the messages may aid with implementation at first, and maintenance of the operational system later.

3.7 Summary

This chapter compared SIP with H.323 in terms of complexity, extensibility, scalability, and services. It was found that SIP provides a similar set of services to those of H.323, but is much less complex; SIP has a rich extensibility and much better scalability. Due to the complex nature of H.323, SIP becomes a more viable alternative for a small company. H.323 seems to be favoured by large telecom type organizations who can invest a lot of manpower in terms of development teams. Once SIP to H.323 interworking solutions becomes available the reasons for using H.323 over SIP will be few. The next chapter addresses the VoIP network design, and the operation of all the elements in the network.

Chapter 4 Design of the Voice over Internet Protocol Network

4.1 Introduction

The protocols governing the transmission of voice over a network were discussed in the previous chapters and it was shown that the transmission of voice and in particular, the method of providing traditional public switched telephone network (PSTN) functionality is a daunting task. The protocols that are currently the standards in voice over Internet protocol (VoIP) telephony, namely H.323 and SIP, were explained and compared. Internet telephony can be defined as telephonic calls over the Internet. These calls can originate from traditional telephone instruments using gateways, personal computers using software or embedded devices also known as Ethernet phones. The greatest motivators for the use of Internet telephony are cost savings and the possibility of the creation and development of new services. The integration of a variety of services provided by the current Internet and public switched telephone network (PSTN) are now possible. The SIP network design and operation will be explained in this chapter.

4.2 Session initiation protocol network components

The architecture of SIP is based on a client/server model, where the client initiates the call and the server answers the call. SIP is a widely supported protocol that does not rely on equipment from specific vendors or implementations. The following paragraphs explain the components that are available for building a SIP network, as well as the principles and operations of the individual components. These components are shown in figure 3.

A SIP network is composed of four types of logical entities. Each entity has a specific function and participates in SIP communication as a client which initiates requests, or/and a server that responds to requests from clients. One physical device, such as the SIP terminal in figure 3 can have the functionality of more than one logical entity, since the client entity will initiate a request and the server entity may respond to a request from another client entity. The four logical entities are:

- User agent (UA).
- Proxy server.
- Redirect server.
- Registrar or location server.

4.2.1 User agent

A SIP UA is an end point entity. UA's initiate and terminate sessions by exchanging requests and responses. The SIP standard defines the UA as application software which contains both a user agent client (UAC) that initiates SIP requests on behalf of the user, and user agent server (UAS) that contacts the user when a SIP request is received, and that returns a response on behalf of the user (Fong 2002:146).

PC's, IP phones, telephony gateways and the SIP terminal in figure 3, are examples of devices that can have a UA function in a SIP network.

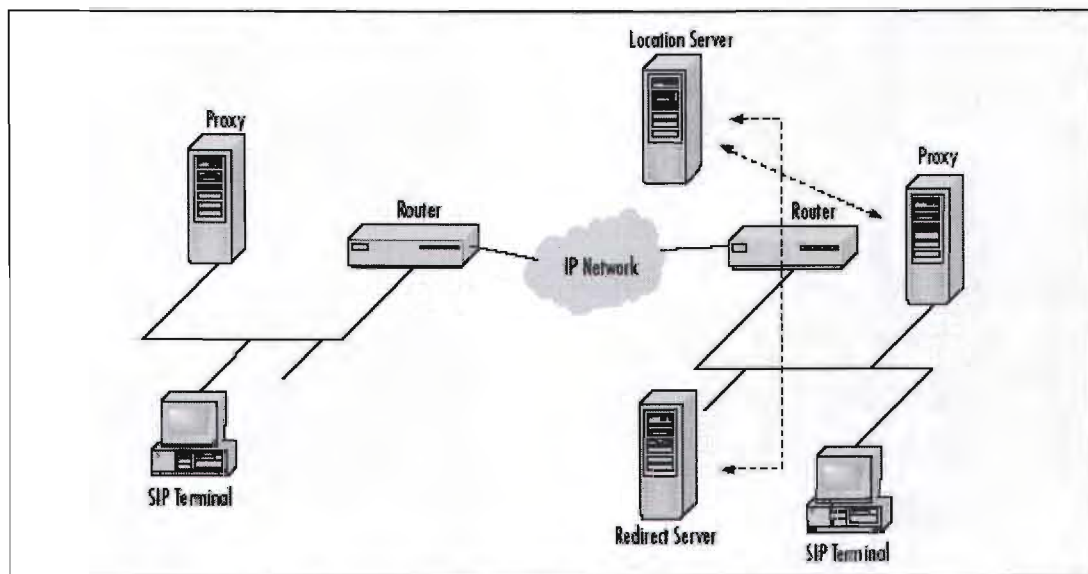


Figure 3: SIP network components (Fong 2002:147).

4.2.2 Proxy server

The proxy server is an entity that acts as both a client and a server. The proxy server decides to which server a request should be forwarded and then forwards it. Such a request can traverse several SIP servers before arriving at the destination. The response follows the same route back (Fong 2002:146). At this time it can be stated that SIP proxy

servers are SIP level routers that forward SIP requests and responses (RADCOM 2004:10). The standard allows for proxies to perform various actions such as validate requests, authenticate users, fork requests, resolve addresses, cancel pending calls, record-route and loose-route, and detect and handle loops. This versatility allows system administrators to use the proxies for different purposes and in different locations in the network. SIP proxies can also be used as management and control devices, for example, to route calls only of authenticated users that do not have outstanding debts to the service provider. Proxies can be placed at the network of the service provider or at the enterprise or small office home office (SOHO) premises (RADCOM 2004:10).

The design of the proxy server is such that it is mostly transparent to user agents, and is only allowed to change messages in specific and limited ways. Also, proxies cannot generate requests at their own initiative, which means that a proxy cannot terminate an existing call by generating a BYE request. The SIP specification defines two types of proxies' namely stateful and stateless proxies. The stateless proxy is a simple message forwarder, which means that when receiving the request the stateless proxy processes the request much like the stateful proxy, however the stateless proxy forwards the message in a stateless fashion, which means that no records are being kept of the message (RADCOM 2004:11). Stateless forwarding allows for improved performance and scalability, but with some consequences which are:

- The proxy server does not retain records of requests it has forwarded, and so cannot associate responses with forwarded requests. Thus the proxy application does not know if a transaction was successful or not.
- A stateless proxy processes retransmissions as if it is the first copy of the message received, and it, therefore, cannot associate retransmissions of requests and responses with previous instances of this message.
- Lost messages will not be retransmitted, since retransmission is the responsibility of stateful user agents or stateful proxies.

Since stateless proxy servers have a high throughput capability, they are often used at the core of carrier and service provider networks, assisting in forwarding SIP messages on

the network. These types of proxies may also be used as load balancers (RADCOM 2004:11). A stateful proxy server processes transactions rather than individual messages. This type of proxy manages two types of transactions, server transactions which receive requests and return responses, and client transactions that send requests and receive responses. As shown in figure 4, an incoming request is processed by a server transaction and then forwarded by one or more client transactions. An incoming response, on the other hand, is received by client transactions and then forwarded back to the server transaction. The association between client and server transactions and the management of the overall state of the request is the responsibility of the proxy core object. Destination addresses are chosen by the proxy core object that initiates one or more client transaction objects accordingly. The proxy core object also collects responses from different client transactions and chooses the responses that will be sent via the server transaction (RADCOM 2004:12).

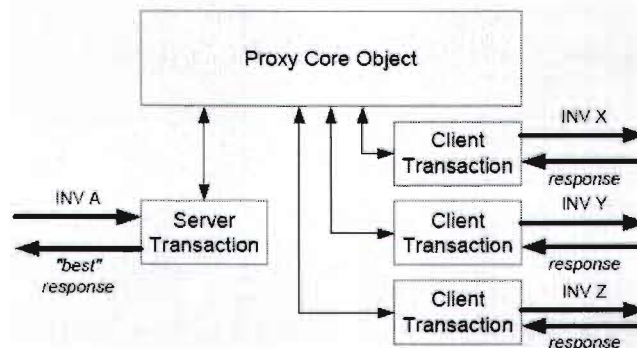


Figure 4: A stateful proxy model (RADCOM 2004:12).

Since a stateful proxy is aware of the state of transactions and message history, it can perform better informed processing on incoming messages. A retransmission of an incoming message can be identified and it can be forwarded only if the situation requires forwarding. A stateful proxy can also generate retransmissions in cases of message loss. A stateful proxy can also process incoming CANCEL requests, and can even generate CANCEL requests when needed. Message forking is also a more natural process for a stateful proxy (RADCOM 2004:12). Stateful proxy servers have a number of disadvantages which are:

- Memory consumption. More memory must be allocated per processed message and for a longer time compared to a stateless proxy. This has a negative impact on the maximum capacity of the proxy and places a limitation on the maximum number of concurrent calls that can be handled (RADCOM 2004:13).
- Throughput. More central processing unit (CPU) cycles have to be spent on message processing, thus reducing proxy capacity and performance (RADCOM 2004:13).
- Implementation complexity. Since the stateful proxy does more than just forwarding requests, logic needs to be employed deal with actions such as parallel forking (RADCOM 2004:13).
- Underlying SIP stack complexity. A proxy requires certain flexibility from the underlying SIP stack, which a user agent does not require. This is especially important at the transport and transaction layers. Therefore, the SIP stack is expected to be more modular and to export more layers of application programming interface (API), compared to a user agent oriented stack.
- Request validation. Before a request can be routed, a SIP server (proxy or redirect) needs to validate the request to make sure it can proceed with message processing. A number of validity checks have to be passed by the message (RADCOM 2004:14).

These checks are:

Reasonable syntax check. Specific fields in the message which the server must process should be checked, all other parts should be ignored or not fixed by the proxy.

URI scheme check. The URI scheme must be a URI scheme the proxy understands and knows how to route. If this is not true, the proxy must return with a 416 (unsupported URI scheme) response.

Max-Forwards check. This is a message field that identifies the number of hops a message is allowed to traverse. For each proxy, one hop decreases this number by one. If

the message contains a Max-Forwards value of zero, the proxy must return with a 483 (too many hops) response. This method prevents a message from going into an endless loop, between a set of proxies.

Loop Detection. The proxy must check that it did not previously handle the message, by executing a loop detection algorithm on the Via list contained in the message. If the message was handled before, it verifies that the message contains different values in the fields that influence routing decisions. If a loop condition is detected, the message is rejected with a 482 response.

Proxy – Require. The client may indicate certain SIP extensions that the proxy must support in order to successfully handle the request. The proxy has to inspect this field and verify that it supports all the extensions listed in the field (RADCOM 2004:14)

Authentication. If the originator of a message has to be authenticated, the SIP server has to make sure the message contains credentials that will authenticate the user. If the message does not contain credentials, or the credentials failed to authenticate the user, the proxy will return a 407 response containing a challenge (RADCOM 2004:15).

Address Resolution. Once an incoming request has been validated by proxy server and it was decided to forward it, the destination/s to which the message is to be forwarded must be determined before sending the message (RADCOM 2004:15).

Two types of address resolution can be performed by the proxy:

Determining the target set. The proxy resolves the request URI to a set of IP addresses that are related to it in some way.

DNS resolution. Each of the destination SIP addresses is resolved by the proxy server to a transport address in the form [transport_protocol, IP address, port].

- **Determination of the target-set.** The first process in address resolution results in producing a set of SIP addresses. This stage maps from SIP address to SIP addresses and is known as obtaining a target-set. A target set is obtained in one of two ways:
Predefined target-set. This is the case where the destination address of the request is such that the proxy must automatically forward to the destination address without trying to resolve to other addresses. A case in point being where the request-URI is in a domain for which the SIP server is not responsible (RADCOM 2004:15)

Target-set determined by proxy. When the target-set is not dictated by the message, the proxy may use whatever mechanism it may wish to determine the target-set. Some options that may be employed are:

- Using a location service updated by a SIP registrar.
- Reading of address information from a database.
- Consulting a presence server.
- Using other protocols.
- Performing algorithmic substitutions on the destination address.

While the Request-URI is an important factor in determining the target set, the proxy may also find a on the basis of other message fields, or on external parameters, such as network and server load or time of day (RADCOM 2004:16).

- **DNS Resolution.** Before a message can be forwarded, the proxy has to resolve the message to concrete transport addresses that can be used in sending the message. SIP entities use the DNS mechanisms described in RFC 3263. This is an algorithm that selectively maps a given SIP address to a prioritized set of transport addresses in the form [transport_protocol, IP_address, port]. The use of this advanced DNS scheme allows for the building of highly available, load balanced SIP networks with the possibility of dynamic adjustment through DNS tables (RADCOM 2004:16).
- **Stateful message forwarding.** The following figures illustrate the way a stateful proxy forwards different types of messages. The term downstream means in the direction of the server, while upstream means in the direction of the client. The scenarios shown show only non-forking proxies that Record-Route (RADCOM 2004:16).

The scenario in figure 5 illustrates the non-INVITE Request-Response message flow through multiple proxies. In the case of non-INVITE requests, such as BYE and REGISTER, the proxy function is to forward requests and responses as they arrive.

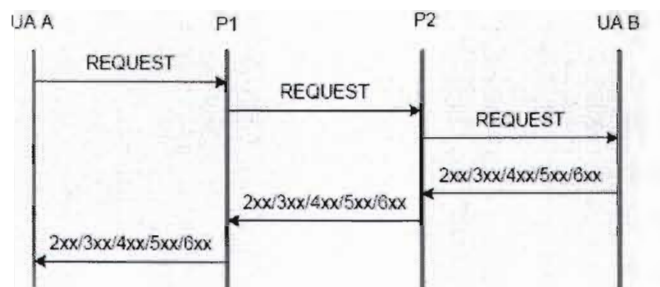


Figure 5: Non-Invite Request-Response message flow (RADCOM 2004:17).

The proxy processes all final responses such as (2xx-6xx) type responses the same way. Retransmitted requests are not forwarded by the proxy. Retransmitted responses are forwarded by the proxy (RADCOM 2004:17). The scenario in figure 6 shows INVITE-non-2xx-ACK message flows through multiple proxies.

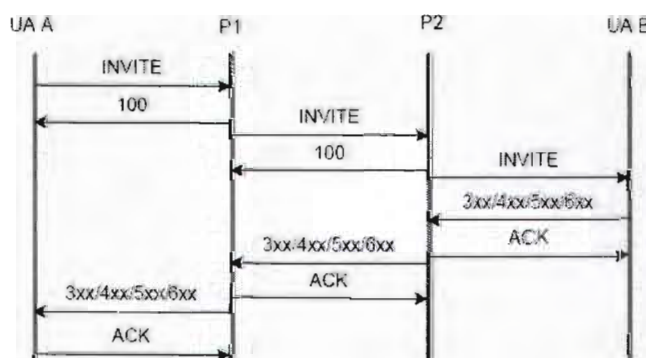


Figure 6: INVITE-non-2xx-ACK message flow (RADCOM 2004:18).

When a proxy is processing an INVITE request, a proxy typically responds with a 100 response to stop INVITE retransmissions. All received 1xx responses with the exception of 100 are forwarded to the previous hop. If the proxy does not receive a 100, it may retransmit the INVITE request when necessary. When a non 2xx response is received, the proxy generates an ACK request and forwards the message upstream. If the proxy receives a retransmission of the request, the proxy retransmits the ACK request (RADCOM 2004:18).

The scenario in figure 7 illustrates INVITE-200-ACK message flow through multiple proxies. A 2xx response to an INVITE request represents a special situation. For the purposes of call setup robustness it is important that reliability of 200 and ACK messages be handled end-to-end, rather than hop-by-hop. This means that when a proxy receives a 2xx response for an INVITE, the proxy forwards the message and any possible retransmission in a stateless fashion. It does not change state in any of the transactions and it does not generate an ACK request (RFC 3261:19).

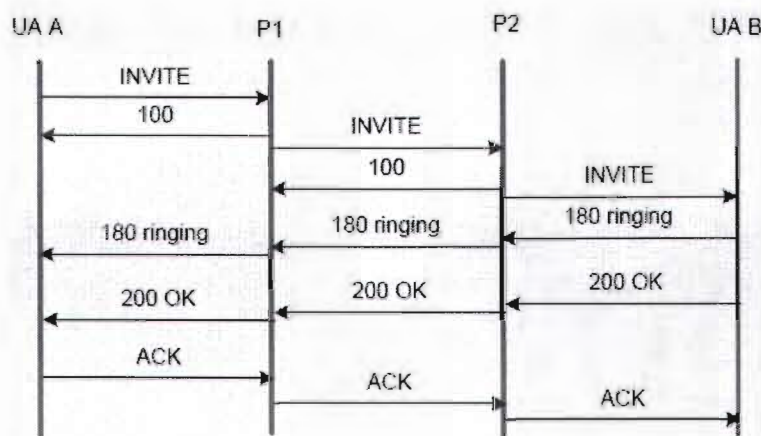


Figure 7: INVITE-200-ACK message flow (RFC 3261:19).

Only UA A in the figure is allowed to ACK a 2xx response. The result is that the proxy will forward the ACK and any possible retransmissions, in stateless fashion. If either the 2xx or the ACK messages get lost, it is the responsibility of the called party, (UA B) to retransmit the 2xx until an ACK is received. This procedure ensures that a call is established and media can start flowing only when both user agents completed the handshake process (RFC 3261:19).

As shown by figure 8, a stateful proxy may generate a CANCEL request for any pending INVITE request that was previously forwarded, subject to the CANCEL rules of the SIP standard. A proxy that receives a CANCEL request must try and match it to an existing INVITE context and cancel any pending client transactions associated with that INVITE. When an INVITE context is not found the CANCEL request must be forwarded statelessly, because the INVITE may have been forwarded statelessly (RFC 3261:20).

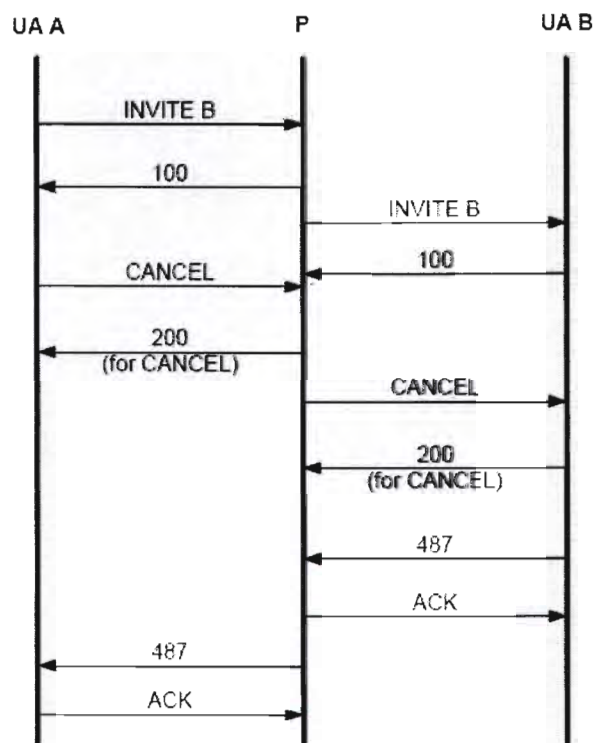


Figure 8: CANCEL processing (RADCOM 2004:20).

Figure 9 shows message proxying without Record-Routing, which is a SIP mechanism that allows a SIP proxy to request to be in the signalling path of all future requests that is part of a particular dialogue.

The process of Record-Routes by a proxy is made possible by entering the Record-Route header into the original request establishing the dialogue. The UAS and the UAC build their route lists based on the Record-Route headers they find in the request send subsequent requests with the route list as a set of Route headers.

When a proxy does not Record-Route an INVITE message, it cannot expect to receive any of the subsequent requests sent as part of the call leg, including the ACK request as can be seen in the figure. Also, a proxy that does no Record-Route a SUBSCRIBE request will not see any of the NOTIFY requests send in the subscription context. Exceptions to these rules may arise from the use of an outbound proxy or from the use of loose routing (RADCOM 2004:22).

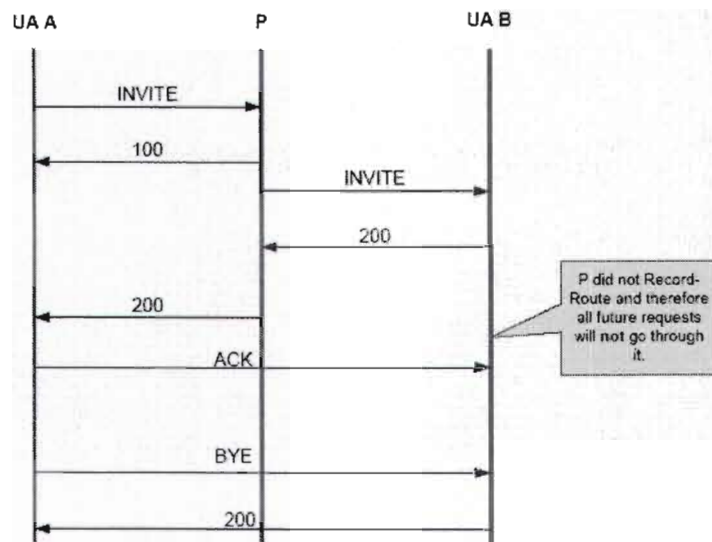


Figure 9: Message proxying without Record-Routing (RADCOM 2004:21).

A proxy Record-Route request is normally used to set up dialogue (INVITE and SUBSCRIBE). Record-Route headers may be added to any SIP request should the proxy wish to do so. User agents only change their route lists based on Record-Route headers of the initial INVITE or SUBSCRIBE. Proxies should avoid adding Record-Route headers to every request so that the processing overhead and message size can be reduced (RADCOM 2004:22).

Selective Record-Routing is very important because a proxy can keep track of some dialogues for the entire duration, and in other cases it can only assist user agents in the initial setup of the dialogue before removing itself from the dialogue. This makes proxy resources available for other tasks. A Record-Routing proxy is required to implement the following functionality:

- Route information pre-processing.
- Route information post-processing.
- Rewrite Record-Route headers in responses.
- Loose Routing.

Loose routing procedures are defined in RFC2543bis-08 and later, and it allows a proxy to introduce more hops into the route-list regardless of the final destination of the message. This allows a proxy to route a message through a pre-defined set of proxies to

its final destination. This allows for intelligent routing of messages between different networks (RADCOM 2004:22).

Figure 10 shows recursion on a 3xx Response whereby a proxy receiving a 3xx (Redirection) response, may add the contact address provided in the 3xx response to the target set, and even generate a new copy of the request to this address. This process is known as recursion on 3xx response, and a proxy may only do that if the Request-URI of the original request indicates an address for which the proxy is responsible (RADCOM 2004:23).

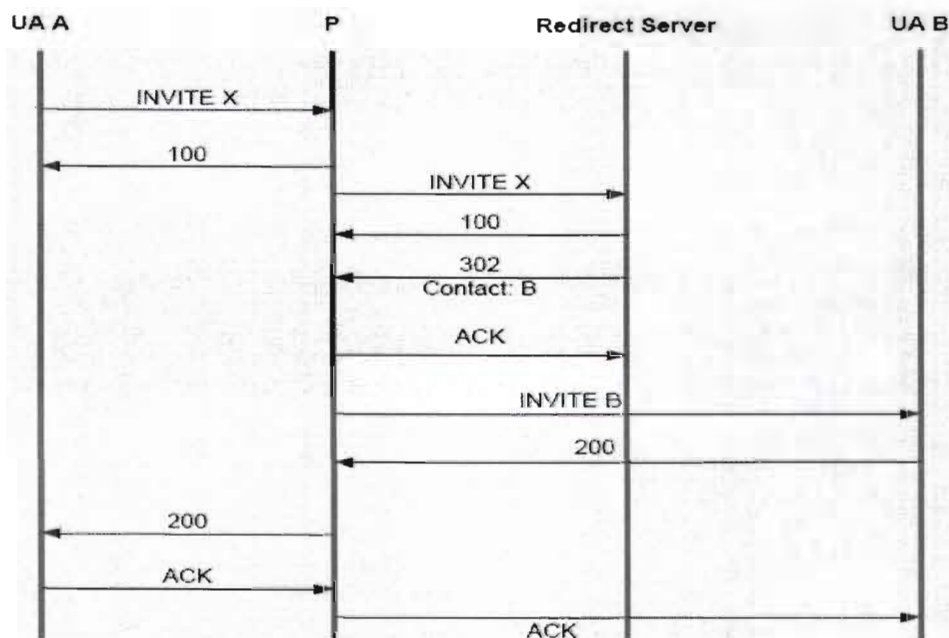


Figure 10: Recursion on 3xx response (RADCOM 2004:23).

After an incoming request has been processed and a target set has been built, the proxy may decide to forward the request to more than one address. This process is known as forking and a proxy that supports it is called a forking proxy. Forking allows the implementation of features such searching for a user. A proxy may use several methods of forking such as:

- Parallel forking: Copies of a request are forwarded to multiple destinations simultaneously.

- Sequential forking: A request is forwarded to one target address at a time, if no response, it will move on to the next address.
- Mixed forking: This is a mix of the above two methods, where some requests are forwarded in parallel and others sequentially.

When this occurs, the proxy has to collect all challenges from all responses and forward them upstream. It may also be required to do redirection response aggregation if multiple 3xx responses are received (RADCOM 2004:26).

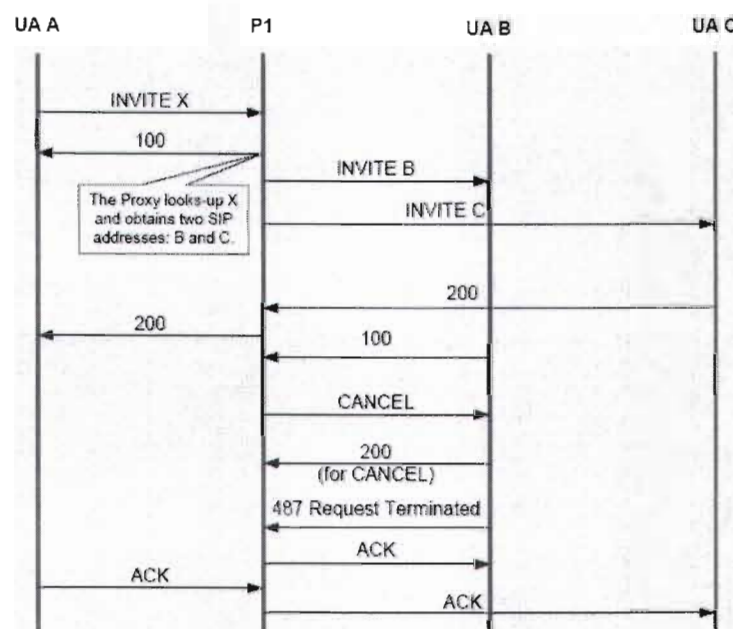


Figure 11: Parallel Forking (RADCOM 2004:25).

Parallel forking as shown by figure 11 is a more efficient way to find a user. This is because parallel forking attempts to find the user at several different locations simultaneously. Parallel forking places additional complexity into the operation of the proxy. The proxy has to handle multiple concurrent client transactions and even possibly collect multiple final responses. The proxy also has to choose the best final response and forward that upstream. The decision for the best response is done according to an algorithm provided in the SIP standard. A parallel forking proxy may also have to do aggregation of challenges if it receives more than one 401 (Authentication required) or 407 (Proxy Authentication Required) responses.

The process of sequential forking of a request is illustrated in figure 12. Here, UA A invites a user that is registered at UA's B and C. User agent B responds with a 404 Not Found message, which then prompts the proxy to send an INVITE to UA C, which then responds with a 200 message, meaning the user was located at this location (RADCOM 2004:26).

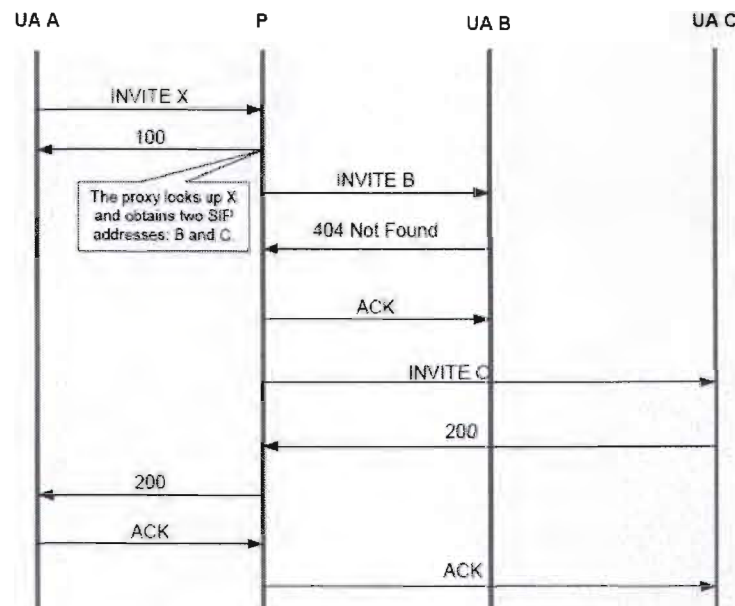


Figure 12: Sequential Forking (RADCOM 2004:26).

The process of authentication is shown in figure 13, where a UAC sends a request to a proxy server, and the proxy server may decide to authenticate the originator before the request is processed. The originator can be challenged by the proxy that returns a 407 response (Proxy Authentication Required) with a Proxy-Authenticate header which contains the challenge. The client can then respond with a Proxy-Authorize header which provides the necessary credentials that matches the challenge.

A client can also provide the credentials before being challenged in order to avoid the delay and extra processing of the 497 response. Authentication by proxy is useful for the following:

- User verification to ensure the user is entitled to receive services.
- Checking that certain message fields was not altered by a third party (RADCOM 2004:28).

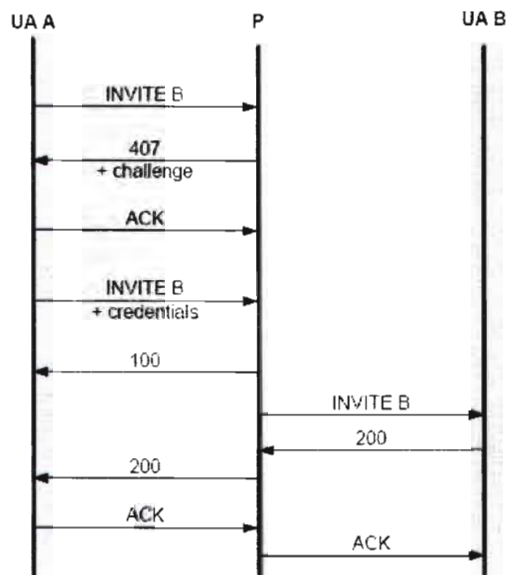


Figure 13: Authentication (RADCOM 2004:27).

A loop is a situation where a request that arrives at a proxy is forwarded enough times so that it arrives back at the origin. As shown by figure 14, an illegal loop occurs if the message arrives at the proxy for the second time, and the message contains the same values in fields that affect the routing decision (RADCOM 2004:28).

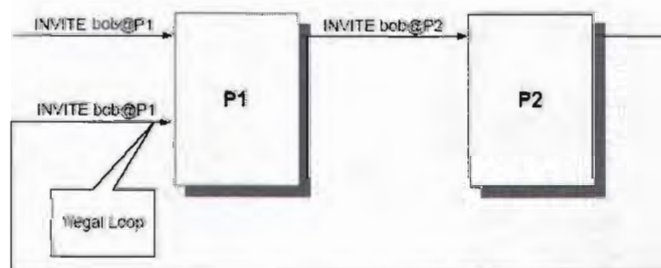


Figure 14: Illegal loop (RADCOM 2004:28).

Loops are handled by the SIP specification in two ways:

- Max Forwards (mandatory).
- Loop detection (optional).

The Max-Forwards header field is used to limit the number of hops a request can pass, on its way to its destination. An integer is used that is decremented by one at each hop. If this value reaches 0 before the destination is reached, it will be rejected with a 483 (Too Many Hops) error response (RADCOM 2004:29).

Optionally, proxies can check for loops by employing a special loop detection algorithm. This algorithm affects the way the proxy builds the Via-branch field and allows the proxy to do certain validations of the Via list in incoming requests. Loop detection places an extra processing burden per message but guarantees immediate loop detection (RADCOM 2004:29).

Message spiralling or legal loops is a SIP request that is routed to a proxy, forwarded and arrives back at the originating proxy but with different sets of values in the fields that affect routing decisions.

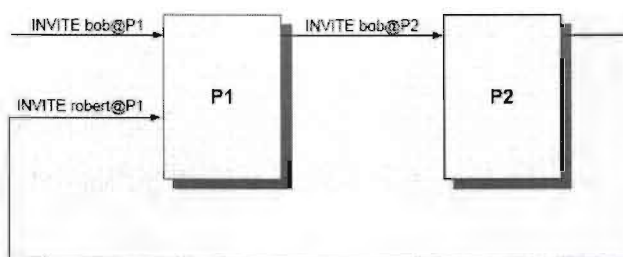


Figure 15: Spiral example (RADCOM 2004:29).

Figure 15 show that the proxy will route the spiralled request to different target addresses the first and second time it processes it. Thus an endless loop is not created. Identifying spirals and distinguishing them from loops presents great challenges for loop detection proxies and requires the use of special techniques (RADCOM 2004:29).

The function of the outbound proxy is that it receives requests from clients regardless of the destination of the message the client is sending. The importance of outbound proxies lies in the fact that they allow for the creation of simpler user agents that do not have to be concerned with making routing decisions and DNS queries (RADCOM 2004:30).

4.2.3 Redirect server

A redirect server is different from a proxy server in that it does not forward requests to other servers. The redirect server notifies the calling party of the location of the destination (Fong 2002:146). It allows proxy servers to direct SIP session invitations to external domains (Furtado 2003:3). This server functionality is the simplest of the three

functionalities. A redirect server receives SIP requests and responds with 3xx redirection responses instructing the client to contact an alternate set of SIP addresses. The alternate addresses are returned as Contact headers in the response message. Table 3 shows the 3xx responses that are currently defined by the SIP standard (RADCOM 2004:7).

Table 4: SIP 3xx responses (RADCOM 2004:8).

Response	Meaning
300 Multiple choices	The address in the request was resolved to several choices, each with its own specific location, and the user can select a preferred communication endpoint and redirect his/her request to that location. This status response is appropriate if the called party can be reached at several different locations and the server cannot or prefers not to proxy the request. NOTE 302 and 302 responses can also contain multiple Contact addresses. The difference is that they convey a more specific reason for the redirection.
301 Moved permanently	The user can no longer be found at the address specified in the Request-URI (the destination address in the request), and the requesting client should retry at the new address given by the Contact header field.
302 Moved temporarily	The user is temporarily available at a different address. The duration of validity of these addresses may be expressed in the Contact header.
305 Use proxy	The request destination address must be accessed through the proxy specified in the Contact field.
380 Alternative service	The call was not successful but alternative services are possible. The alternative services are described in the message body of the response. The use of this response code is still not defined in SIP and is intended for future use.

It will be found that in most cases 301 and 302 responses are used by redirect servers, although 300 can also be used, but it is more ambiguous to the calling client (RADCOM 2004:8).

The example in figure 16 illustrates a redirection scenario, where a user at client A invites user X for a conversation. This request is sent to the redirection server, which responds with a 302 moved temporarily message. Thus client A acknowledges this and takes note of the new address at which user X can be contacted and proceeds to do so. User X's client responds with a 200 OK message which is acknowledged by user A's client.

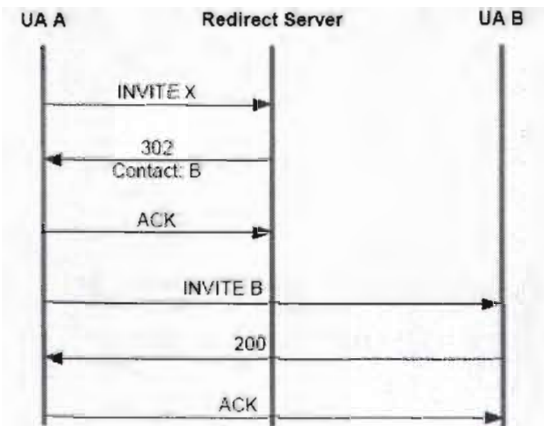


Figure 16: The request redirection process (RADCOM 2004:9).

It must be noted that the second invite request is generated with the same dialogue identifiers, Call-ID, and To and From headers as the first INVITE request, but with different CSeq values. Redirection servers push back routing information for a request in a response to the client, with the result that it aids in locating the target for the request, while not being involved in further messaging for the transaction. Redirect servers are typically unaware of the state of dialogues, but only of the state of the individual transactions they are handling, making them transaction-stateful elements (RADCOM 2004:9).

4.2.4 Registrar or location servers

These servers provide registration services for UAC's. They contain databases with the locations of all UA's within a domain. These servers retrieve and send participants IP addresses and other information to the SIP proxy server (Furtado 2003:3). Registration servers are often contained within the same hardware as proxy and redirect servers (Fong 2002:146).

As shown by figure 17, a registrar server can be defined as a server that accepts REGISTER requests and places the information received in those requests into the location service for the domain it handles. REGISTER requests are generated by the clients in order to publish or remove the relationship between their externally known SIP addresses and the addresses they wish to be contacted at (RADCOM. 2004:5).

The registrar processes REGISTER requests only for a specific set of domains and it uses a location database to store and retrieve location information. This location service may run on a remote computer and it may be contacted by using an appropriate protocol such as lightweight directory access protocol (LDAP). All incoming REGISTER requests may be authenticated using the 401 response (RADCOM 2004:6).

It must be stated that most often, more than one of the above logical entities are combined into one physical implementation, such as a SIP server that may have location and registration functionality as part of its implementation.

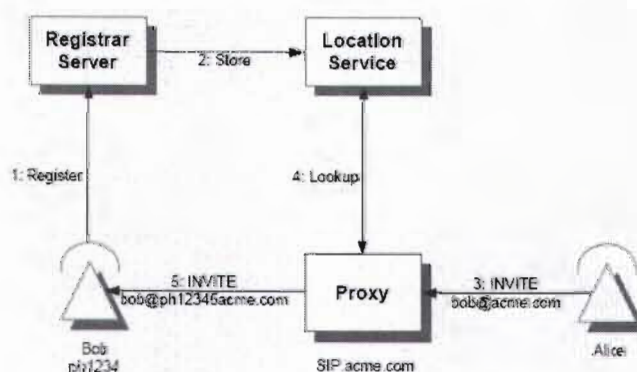


Figure 17: SIP Registration Process (RADCOM 2004:7).

4.3 Detailed session initiation protocol message flow

The next step in understanding the SIP network operation is the sequence of events that take place when users call each other, and this is discussed next.

A SIP transaction is depicted in figure 18, showing the flow of events when a user places a call to another user. The SIP user agent creates an INVITE request for

sip:jakes@vut.ac.za; this request is forwarded to a local proxy (1). This proxy looks up vut.ac.za in DNS, and obtains the IP address of the server handling SIP requests for this domain, it then proxies the request to this server (2). The server for vut.ac.za knows about the user Jakes, but this user is currently logged in as j.user@university.edu. So, the server redirects the proxy (3) to try this address. The local proxy looks up university.edu in DNS, and obtains the IP address of its SIP server. The request is then proxied there (4). The university server consults a local database (5), which indicates (6) that j.user@university.edu is known locally as j.smith@cs.university.edu, so, the main university server proxies the request to the computer science server (7). This server knows the IP address where the user is currently logged in, so it proxies the request there (8). The user accepts the call, and the response is returned through the proxy chain back to the caller (Schulzrinne 1999:4).

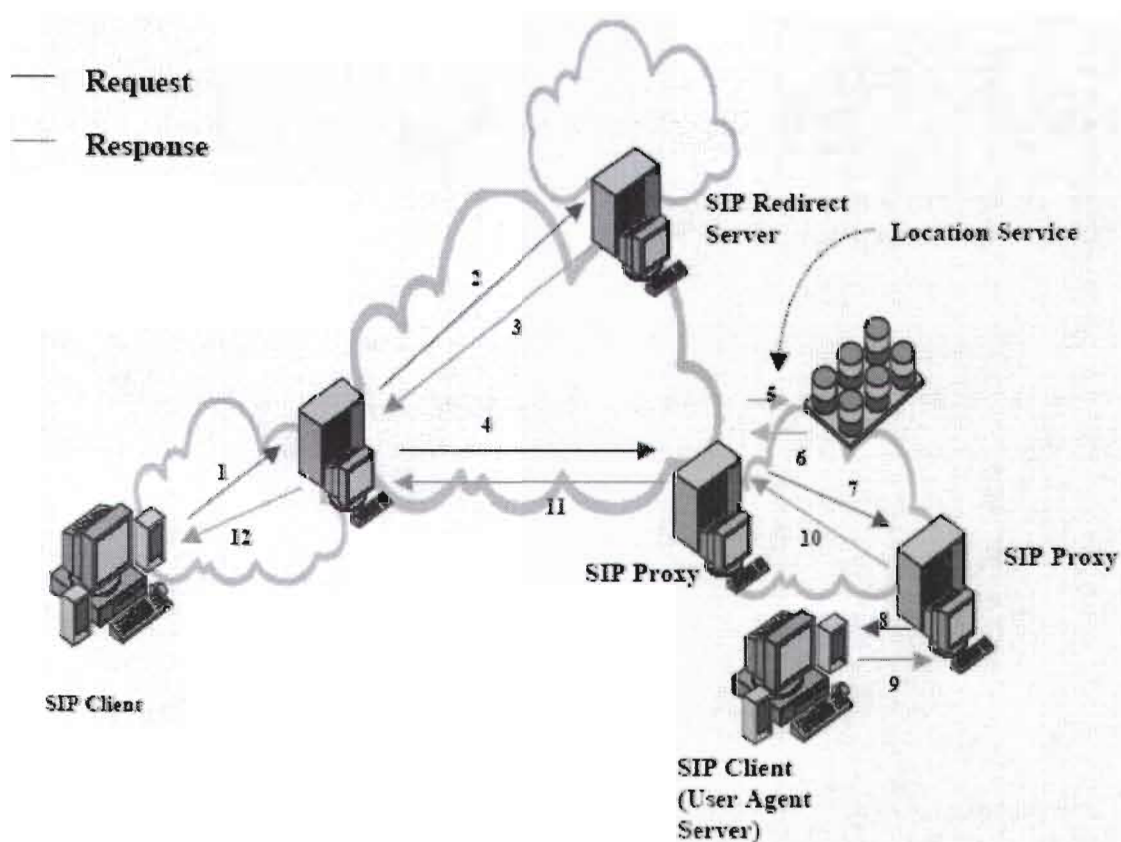


Figure 18: Detailed SIP message flow (Schulzrinne 1999:4).

4.4 The network design and test environment

Now that the operation of the SIP network with all its components is understood, the detailed designed network that was used as the actual environment is discussed with the use of figure 19. UA software was installed on desktop 1 and desktop 2 computers. The stateful proxy server software with its specifications and features shown in annexure 4 was installed on a notebook computer. The users that are allowed to use the service were created in the proxy server database.

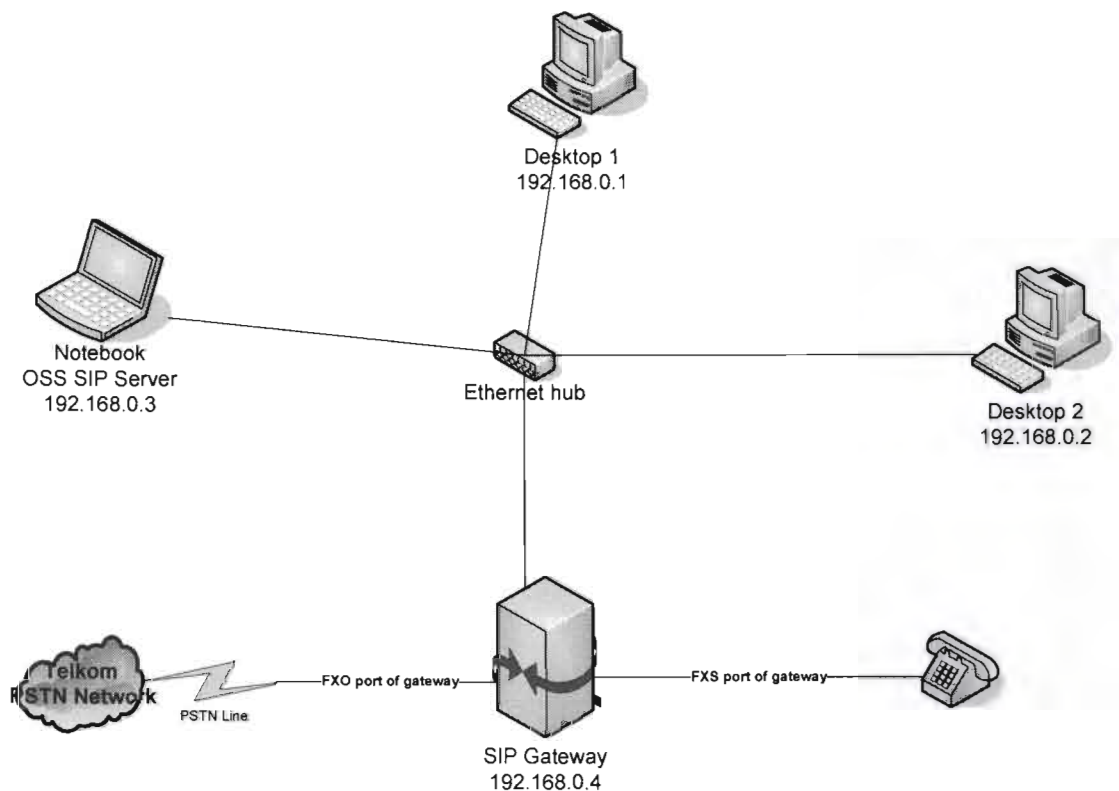


Figure 19: The primary network design and test environment.

This network operation is as follows: Whenever desktop 1 and 2 are started the soft phone software would automatically start searching for a proxy server where it could register. Once the registration process is complete, users at desktop 1 and 2 can now call each other. The key being that a user can call any other user whose soft phone is successfully registered on the proxy server. Suppose user at desktop 1 registered as Joe and user at desktop 2 registered as Pete, user Joe can call user Pete by using the URL Pete@192.168.0.3. Although user Pete is actually located at address 192.168.0.2, the proxy makes use of the location feature of the SIP protocol to establish a connection. The

purpose of the gateway depicted in figure 19 is to allow a user to call a regular PSTN number from his desktop computer. The gateway specifications and features are listed in annexure 4. The process here is different in that the foreign exchange office (FXO) and foreign exchange station (FXS) ports of the gateway register on the proxy server as services being available for use. Therefore when user Pete wants to dial the FXS number 7701, all that is needed is he dials the number using his soft phone.

The next question that arises is how a user at the telephone connected to the FXS port can dial a user's soft phone. In order to accomplish this, a change is needed in the user data base. Instead of registering users by their names as indicated earlier, it is necessary to register users using a number for instance 7702, 7703 and so on. The users will still be called in the format 7702@192.168.0.3 if being called from a soft phone, however, should the user be called from the telephone at the FXS port, the user can simply be called by the user number, 7702 for example.

The next problem to be solved is how a user dials a regular telephone number using his soft phone. In order to accomplish this, a dial plan must be created on the proxy server. The purpose of the dial plan is to take a request for a service that is not available on the network and route it to an entity that does know about the service. The proxy server has a dial plan configuration option that allows an administrator to implement dial plan configurations. The implemented dial plan is shown in annexure 1 and will be explained here.

In Matching Patterns, conditions for call session control settings can be set, and all variables start with a \$ symbol. Deploy patterns define deploy actions for cases where the condition is fulfilled in Matching patterns. Thus, in the example, if the server receives an INVITE request that starts with a 0 and contains 10 digits, such a call will be routed to the gateway; (in this case the IP address of the gateway) such as shown by the Deploy pattern in annexure 1.

4.5 Design implementation

The network in figure 20 shows a typical departmental or satellite campus network at VUT. This network meets with the VUT campus network at the 4507 Cisco core switch if it is a departmental network or via the Cisco 2600 router if it is a satellite campus network. It must be pointed out that only UA software was installed on users desktop PC's in these networks. Another important factor was to make sure that the UA software registered correctly on the SIP server which is located at the departmental network (Figure 21).

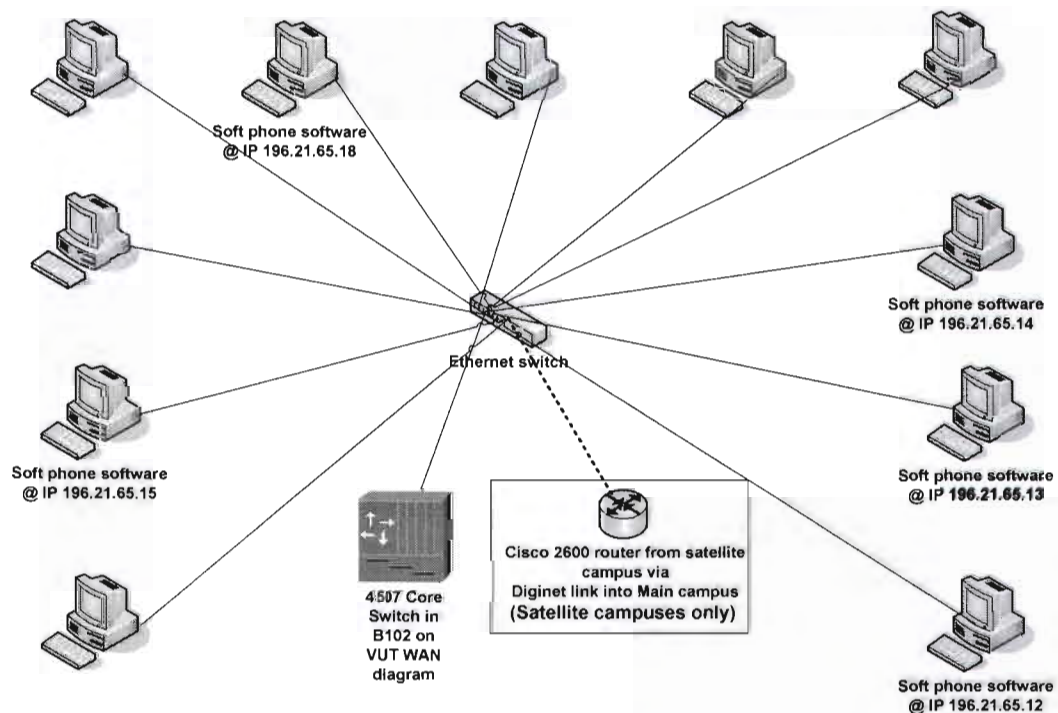


Figure 20: Departmental or satellite campus network.

The departmental network which was used as the implementation environment of this design is shown in figure 21. This network is essentially the same as figure 20, however it will be noted that there are a couple of extra hardware devices. The first is the proxy server, and the second the SIP gateway. Since the network of VUT is a production network that is being used on a daily basis, this network is located behind a firewall in order to protect the users and property of VUT from malicious attacks from the public Internet. This has the effect that all traffic that is perceived as potential threats is blocked

by the firewall. Keeping this limitation in mind during the design process, it became evident that an alternate location had to be found for the SIP server, and gateway. Permission was obtained to use this network, since it provided a secure and stable environment and it had access to an open exchange line to the Telkom PSTN, which was needed for testing off network call situations.

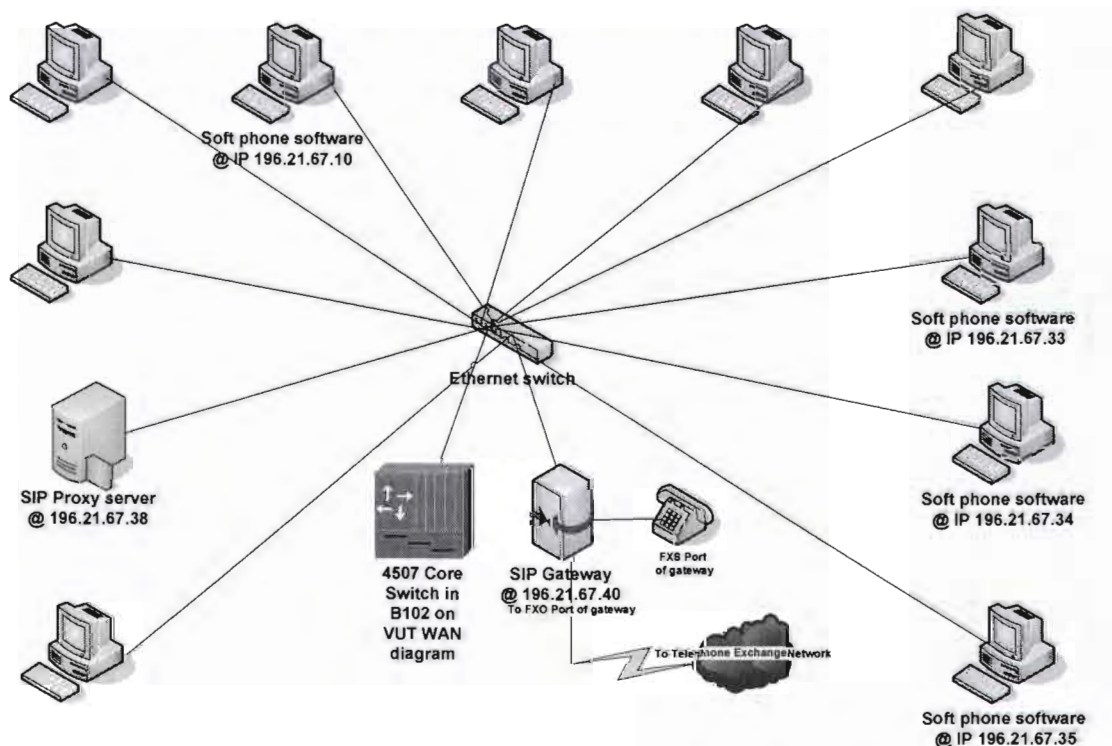


Figure 21: Design implementation environment.

The basis of operation of the SIP network is the same as the test network environment that was explained earlier.

4.6 Convergence at the Vaal University of Technology network

The departmental and satellite campus networks converge as the entire network of VUT (Figure 22). The VUT network is a switched network, with a number of switches placed at strategic places throughout the campus. The reason for using switches is to divide the large University network into smaller contention domains, which theoretically should improve the overall network performance. Functionally switches are being used to reduce unwanted traffic while broadcasts are being flooded (Kutz 2002:149).

As shown in figure 22 a number of Cisco routers provide access to the main campus network. Port forwarding must be implemented on the routers so that the soft phones at the satellite campuses can register with the proxy server. The manufacturer of the soft phone that was used in this research states that "port forwarding" for the following ports UDP 5060, 5062, 30000-30021 should be implemented. This was the case on the University network, and therefore did not present any problems. This design and its implementation are in use on a daily basis, and are already accepted as part of the communications infrastructure of the VUT.

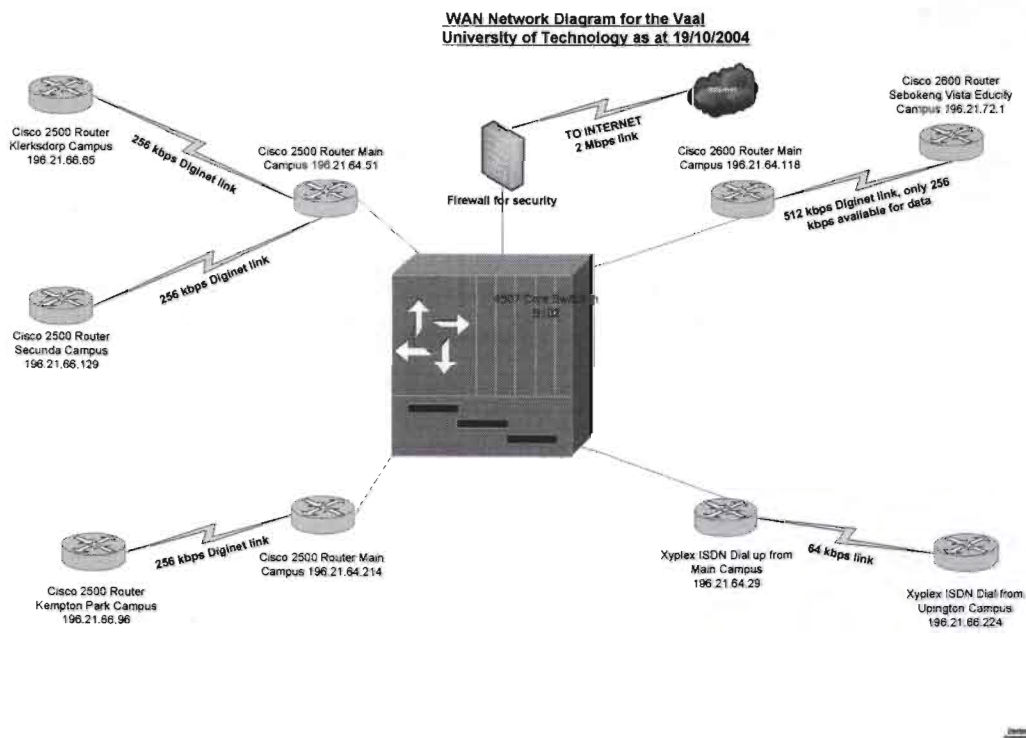


Figure 22: The Vaal University of Technology Wide Area Network (WAN).

Shown in annexure 2 is an extract of the call log of the SIP server that was implemented during this research. This log was used for management and administration purposes. A short explanation of the extract from the call log may aid in the understanding of the events. User 57521vanderschyff placed a call to user Dewald on Tuesday October 12 at 10h26. The call duration was from 10h26 until 10h29 (a total call time of 3min 12 seconds) and the call was completed successfully.

4.7 Summary

The hardware components of a SIP network and their functions were described in this chapter. The basic network that was used to prove this design was described along with its operation. Next it was shown how a basic SIP network design was integrated into the working network of the VUT. The next chapter addresses bandwidth measurements.

Chapter 5 Measurements and results

5.1 Introduction

The first question that comes to mind when reading about VoIP is how much bandwidth is consumed by an audio stream. To answer this question there are a number of questions that need to be answered. The first factor is the codecs used to code and decode the analogue audio into a digital format. Second is the bandwidth required by the transmission medium overhead, where the amount of bandwidth required by the transmission protocol and its components must be considered. This chapter will address all of the above issues and it will show that in today's local area data networks, bandwidth availability is one of the most abundant resources available and that calculated and measured values compare very well.

5.2 Review of relevant measurement theory

As long ago as 1924, H.Nyquist realized that a data channel must have a limitation, and derived an equation expressing the maximum data rate for a finite bandwidth noiseless channel. In 1948 Claude Shannon continued with this work and extended it to a case of a channel that may be subject to random thermal noise. Nyquist proved that for an arbitrary signal passed through a low pass filter of bandwidth H , the filtered signal can be reconstructed by making only $2H$ samples per second. It was also found that sampling faster than $2H$ times per second is pointless, because the higher frequency components that such sampling could recover have already been filtered out. Furthermore if the signal consists of V discrete levels, Nyquist's theorem states that,

$$\text{Maximum data rate} = 2H \log_2 V \text{ bits per second}$$

This means that a noiseless 3 kHz channel cannot transmit binary signals at rates exceeding 6 kbps (Tanenbaum 1989:56).

Until now, only noiseless channels have been considered. The moment random noise is present, the situation deteriorates rapidly. The amount of thermal noise present is measured as the ratio of signal power to noise power, also known as the signal to noise

ratio. If the signal power is represented by S , and the noise power by N , the signal to noise ratio can be expressed by S/N . Shannon therefore, produced a formula that could be used to calculate the maximum data rate of a noisy channel whose bandwidth is H , with a signal-to-noise ratio of S/N and this formula states that:

$$\text{Maximum number of bits per second} = H \log_2 (1+S/N)$$

For example, a channel with a 3 kHz bandwidth and a signal to thermal noise ratio of 30dB can never transmit more than 15 kbps. These results of Shannon were derived using information-theory arguments and have very general validity (Tanenbaum 1989:57).

5.3 Voice over Internet protocol bandwidth calculation

To answer the question of how much bandwidth a VoIP call occupies, a number of factors need to be considered. The most important of these factors are:

- Codec (coder/decoder) sample period.
- IP header.
- Transmission medium.
- Silence suppression.

A short explanation will now be given of the function of each of the above factors. The codec being used by the clients during the voice call will determine the actual bandwidth that the voice data will occupy. It is also the determining factor of the rate at which the voice will be sampled. The IP/UDP/RTP header has, as a general rule a fixed overhead of 40 octets per packet that can be reduced to 2 to 4 octets when RTP header compression is used on point-to-point links. The transmission medium such as Ethernet will add its own headers and checksums to the packet. It must also be considered that some codecs employ silence suppression, which can reduce the required bandwidth by as much as 50% (VoIP – MoIP tech note 2004:2).

- The codec.

The codec is responsible for the conversion of the analogue waveform to a digital form. The audio waveform is sampled at regular intervals and generates a digital value for each

sample. A typical sampling rate is 8000 times per second. These individual values are accumulated for a fixed time period in order to create a frame of data. A common sampling period would be 20 ms, the important characteristics of the codec are:

- Number of bits produced per second.
- Sampling period – defines the rate at which samples are transmitted.

The above factors combined determine the size of the frame. As an example a G.711 codec sampling at 20 ms generates 50 frames of data per second. G.711 transmits 64 kbps, therefore each frame will contain $64 \text{ kbps} \times 20 \text{ ms} = 1280 \text{ bits}$ or 160 octets (VoIP – MoIP tech note 2004:2).

- Frames and packets

The IP telephone, depending on its type, can place either only one frame of data in each packet, or more than one frame in each packet. For example, if an IP telephone uses the G.729a codec which works with a 10 ms sample period, and produces a very small frame of 10 bytes only. It can be seen that it will be more efficient to place two frames in each packet, which will decrease the packet transmission overhead without increasing the latency too much (VoIP – MoIP tech note 2004:2).

- Latency and packet overhead.

The numbers of frames per packet is a balancing act between two characteristics namely latency and packet overhead. High latency is the result of long sampling periods which may affect the perceived quality of the call. Based on this fact alone it is evident that shorter sample periods will produce better perceived call quality. However this results in smaller frames with more significant packet headers. For the smallest packets, more than half the bandwidth used is taken up by packet headers, a very undesirable situation since the aim is to use the maximum available bandwidth for audio and not waste it on the transport mechanism (VoIP – MoIP tech note 2004:3).

- The IP header.

The term IP header refers to the combined IP, UDP and RTP information placed in the packet. The voice data generated by the codec is encapsulated in successive layers of

information in order to deliver it to the destination. These layers are:

- IP – Internet protocol.
- UDP – User datagram protocol.
- RTP – Real Time Transport protocol.

RTP is the first layer added, and is 12 octets in size. RTP is used to reconstruct the samples in the correct order and it also provides a means for measuring delay and jitter. UDP adds a further 8 octets and is responsible for routing the data to the correct destination port. UDP is connectionless and does not guarantee delivery or packet sequence information. IP adds 20 octets, and is used to deliver the data to the destination host. IP is also connectionless and also does not guarantee packet sequence information (VoIP – MoIP tech note 2004:3). In total, the IP/UDP/RTP headers add a fixed 40 octets to the payload. With a sample period of 20 ms an additional 16 kbps is added to whatever codec is being used. Consider the G.711 codec discussed earlier again; 20 ms samples produces 160 octets, and the IP header adds 40 octets, resulting in 200 octets, or $1600 \text{ bits} * 50 \text{ times per second} = 80 \text{ kbps}$. This result does not take into account the physical transmission medium bandwidth requirement (VoIP – MoIP tech note 2004:3).

- The transmission medium.

In order to traverse the IP network, the IP packet must be wrapped in another layer by the physical transmission medium. Most IP transmissions start their journey over Ethernet and parts of the core transmission network is also most likely to be Ethernet. The minimum Ethernet payload size is 46 octets. In order to carry IP packets with a fixed IP header of 40 octets means that the codec data must be at least 6 octets. The Ethernet packet starts with an 8 octet preamble followed by a header of 14 octets defining the source and destination media access control (MAC) addresses, and the length. This payload is followed by a 4 octet cyclic redundancy check (CRC). Finally the packets must be separated by a 12 octet gap. Resulting in an additional Ethernet overhead of 38 octets. Therefore Ethernet adds a further 38 octets to the previous 200 octets of the G.711 codec example of earlier. Thus,

Codec G.711 – 64 kbps, 20 ms sample period

1 frame per packet (20 ms)

Standard IP headers

Ethernet transmission medium

One packet transmitted every 20 ms = 50 packets per second.

Payload = 64 kbps / 50 = 1280 bits

Fixed IP overhead = 40 octets or 320 bits

Fixed Ethernet overhead = 38 octets or 304 bits

Total size = 238 octets or 1904 bits

Bandwidth required = $(1280+320+304)*50*8 = 95200$ bps

Which is the minimum bandwidth required to transmit voice over IP over Ethernet using the G.711 codec (VoIP – MoIP tech note 2004:4).

Another example: This time the G.729a codec, with a 10 ms sampling period will be used. Since a packet is sent every 20 ms two samples of 10 ms will be packaged. Thus:

Codec G.729a – 8 kbps, 10 ms sampling period

2 frames per packet

Standard IP headers

Ethernet transmission medium

One packet is sent every 20 ms, which translates to 50 packets per second. The payload is $8000 / 50 = 160$ bits or 20 octets. The fixed IP overhead of 40 octets and fixed Ethernet overhead of 38 octets equals to a total packet size of 98 octets.

Bandwidth required = $(20+40+38)*50*8 = 39200$ bps

- Silence suppression.

Some codecs support silence suppression. Also known as voice activity detection (VAD), which suppresses the transmission of data during silent periods. Since only one person normally speaks at a time, the demand for bandwidth can be reduced by as much as 50 percent. During such silent periods, the receiving codec will normally generate comfort noise (VoIP – MoIP tech note 2004:5). However, silence suppression had the perceived effect of delay during conversations, and from a user perspective, perceived voice quality is much better with silence suppression not active.

5.4 Actual Voice over Internet protocol bandwidth measurements

In order to test the validity of the above calculations, an experiment was set up in order to measure the actual bandwidth consumed by a VoIP call between two clients. The physical transport medium was Ethernet, and in order to get unspoiled data, no other processes were run on the computers that acted as the clients. The physical network media used was a 10 Mbps unshielded twisted pair cable. The actual soft phones were setup to use a specific codec during that specific call as indicated on the graphs.

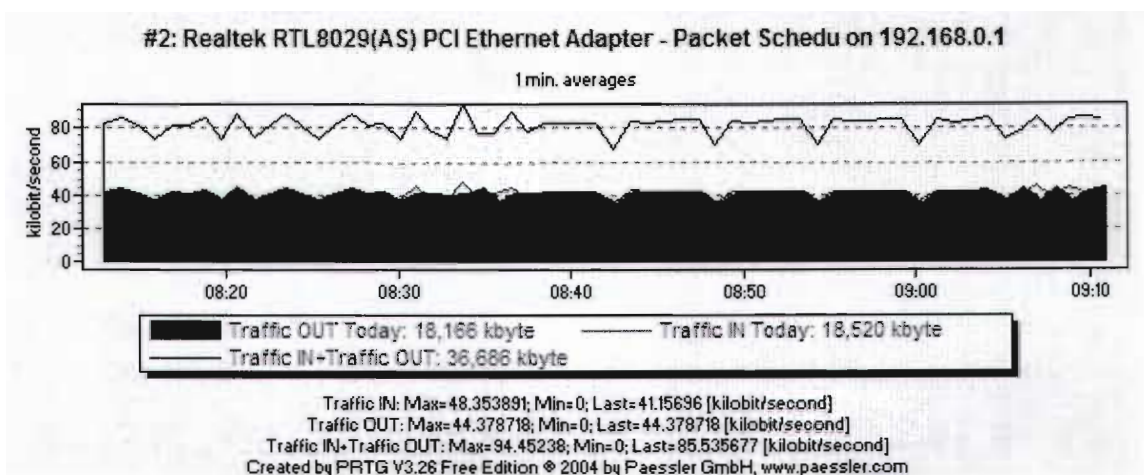


Figure 23: Bandwidth occupied during SPX codec call.

By analyzing figure 23 it can be seen that the average data received was 48,3 kbps and the data transmitted was 44,3 kbps. The total amount of bandwidth needed was an average of 94,4 kbps data in and out of the monitored network adapter. The bandwidth occupied during an iLBC codec call is shown in figure 24 as transmitting 33 packets per second, and here it can be seen that there was a significant decrease in bandwidth used. Traffic in measured 33,4 kbps and traffic out measured 33,0 kbps with a total of 66,4 kbps in and out of the network adapter.

Figure 25 shows the bandwidth used during a GSM codec call that transmits 50 packets per second, and it shows a slight increase in bandwidth used to the iLBC codec but virtually the same as the SPX codec call. Measured traffic amount into the adapter was 42,5 kbps and out of the adapter 41,8 kbps, with a total bandwidth occupation of 84,4 kbps. The graph in figure 26 shows a significantly more occupied bandwidth than all the

other previous cases. However, what should be noted on this graph is that the actual measured results correlates very closely to the G.711 codec calculation that was done earlier. The calculation showed a 95,2 kbps bandwidth use for this particular type of codec, and the actual bandwidth measured was 101,5 kbps.

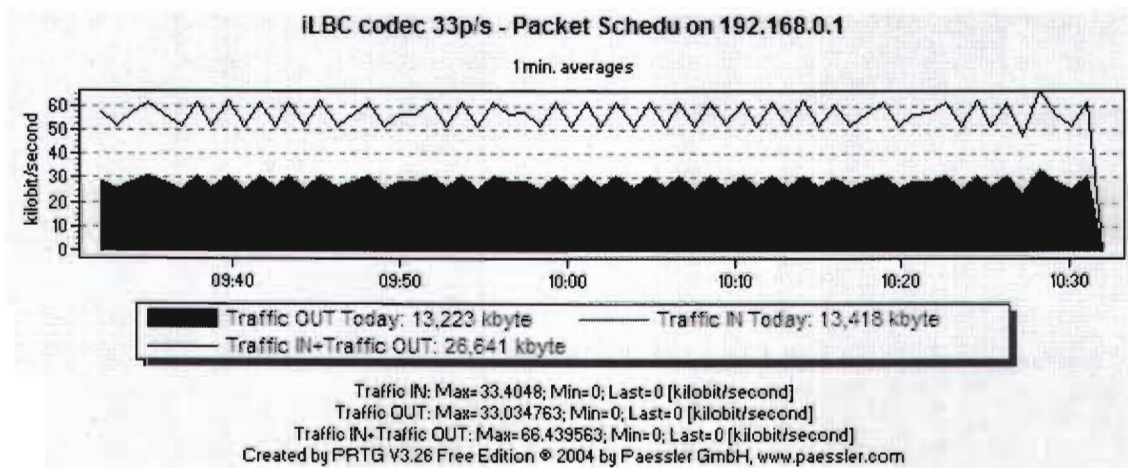


Figure 24: Bandwidth occupied during an iLBC codec call.

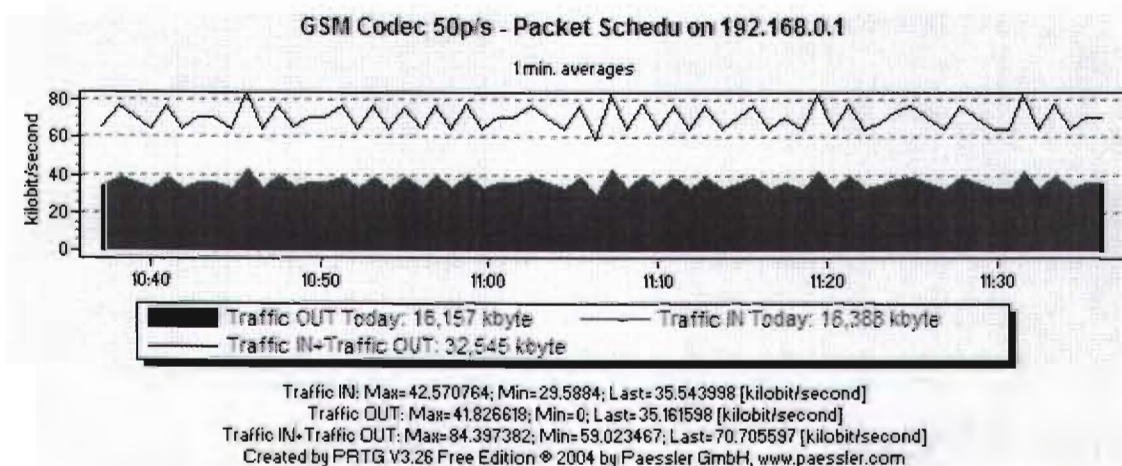


Figure 25: Bandwidth occupied during a GSM codec call.

The bandwidth occupied during a G.711a codec call is similar to that of a G.711u codec call (Figures 26 and 27). There is no significant difference in used bandwidth by either of these two codecs. From the experiments it became evident that the bandwidth used on a particular media can be managed by choosing the appropriate codec type, resulting in bandwidth savings. It must be mentioned that the perceived quality of the voice during all of these measurements stayed the same.

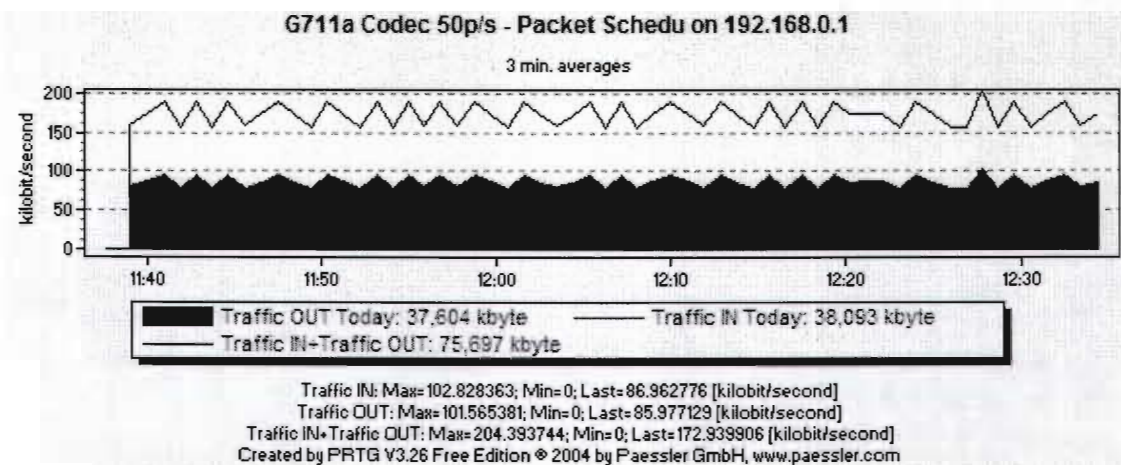


Figure 26: Bandwidth occupied during a G.711a codec call.

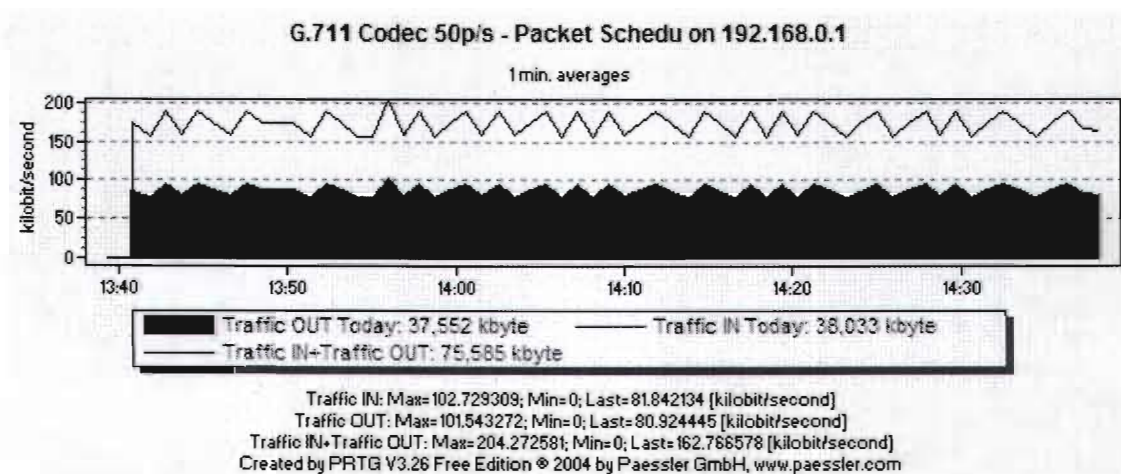


Figure 27: Bandwidth occupied during a G.711u codec call.

5.5 Audio quality results

On a daily basis, calls are being placed from soft phone to soft phone, soft phone to normal telephone, and soft phone to cellular phone on the data network of VUT. The user's reports indicate consistently excellent voice quality during all of these calls. This can be attributed to the bandwidth being available, since the soft phone's negotiates to use the best codec for the available bandwidth. It was found that when the network experienced heavy data loads on slow WAN links, jitter did occur. But at no stage was it so bad that such a call became unintelligible. It was also found that better audio quality was reported when a particular soft phone was used to place calls to the telephone connected to the FXS port of the gateway. The reason for this cannot be explained since

both types of soft phones support similar codecs, and it was decided to use the soft phone that produces the best results to all call destinations.

5.6 Summary

This chapter addressed the question of bandwidth consumption during a VoIP call. The factors that influence the bandwidth usage are the codec being used by the soft phone, the IP header with its fixed overhead, the transmission medium with its overhead and whether or not silence suppression is being used. By approaching the problem one element at a time the final calculations become relatively simple. The measured results show that there is a close correlation between calculated results and experimental results. The users using this system on a daily basis consistently report excellent audio quality during all calls. The next chapter concludes this thesis. The findings and implications are discussed, and a number of recommendations are made.

Chapter 6 Conclusion

6.1 Findings and implications

The results of this study show that the two possible VoIP standards investigated presented a solution each ranging in complexity, expense and proprietary in nature. This resulted in looking for a system that is less intricate, with software and hardware that is readily available, and that could be implemented without the need for state of the art equipment. Two standards were found, H.323 and SIP. H.323 is quite complex and proprietary in nature, and SIP rather simple, with a large Internet community that is prepared to share resources and knowledge. The two standards were evaluated against each other, in order to find the most suitable solution. The decision was made based on criteria such as complexity, scalability and extensibility. The chosen system software was installed on a trial basis on a number of computers, in order to demonstrate the suitability of the chosen standard.

This system uses SIP that comprises three basic entities, namely user agent clients and user agent servers, a proxy, and a redirect or registrar server. The proxy, redirect and registrar servers are most often combined in a single implementation. The system operates on the same principles as HTTP for locating resources. A user's soft phone would register on the server, and once registered, this user can be called in the format `user@ip_address_of_server`. These kinds of calls are calls within the data network, and most of the time the users report outstanding call audio quality.

Should the users wish to place a call outside the data network, a fourth piece of equipment is needed, namely the gateway. The operation of the system then changes slightly because the gateway also registers on the proxy server. A short program called a dial plan must also be entered on the server. The effect of the dial plan is to receive INVITE requests and route them to the gateway in the case where the INVITE request does not contain a request for the services of an entity on the network. Here the user would simply dial a telephone number the usual way, from his soft phone. The system

takes care of the routing of the call from the data network to the public switched telephone network.

The amount of bandwidth consumed during such a call is surprisingly small considering the amount of information being transmitted. Upon closer inspection, it was found that the major consumer of bandwidth is the type of codec being used, the amount of data being transmitted by the codec per second, and the sampling period of the codec.

6.2 Reassessment of the problem

In order to understand voice communication over a data network, it is important to understand the standards that govern VoIP communications, as well as the hardware and software components that will ultimately make up the VoIP system. Such a system was found to be viable since only two major sets of standards govern this type of system. The H.323 standard is proprietary, with a standards document in excess of five hundred pages and the implementation of this system will require a large amount of resources in terms of money and hardware.

The SIP set of standards, is a development within the Internet community, and a large amount of people are busy with its development, resulting in a vibrant, dynamic process of sharing resources and information. After evaluating both standards based on a predefined set of criteria it was decided to use SIP. Another important factor that contributed to this was the fact that the software required is available for free and the support from the software developers is second to none.

A SIP system was ultimately deployed in a department that does not have access to traditional telephony services, and the head of the department expressed a need to be able to communicate with the people within the department. This progressed further into a system where the users within the department are able to place calls to other departments and even satellite campuses using the data network, and calls can also be placed into the public switched telephone network, from the data network.

It was shown during this research that it is possible to use the data network as a reliable network for the transmission of voice with call quality equal to, if not better than the traditional telephone network. The impact, if any, of the voice traffic on the data network was negligible.

6.3 Recommendations

In the context of the Vaal University of Technology a number of factors could contribute towards the successful deployment of this system on a University wide basis, thus eliminating the need for a PABX telephone system entirely with its associated cost in terms of capital and expenditure.

- Firstly, the wide area links between the main campus and the satellite campuses is inadequate. At the Educity campus up to eighty users are making use of the e-learning resource, which translates to 256 kbps per 80 users = 3,2 kbps per user available bandwidth, this excludes any other users which may be accessing their e-mail or doing research on the Internet at the same time, which will decrease the available bandwidth even more. Should a VoIP system be deployed on this already limited bandwidth a significant decrease in performance and call quality will be experienced not to even mention the decrease in Internet search times and e-mail download performance. This is also true for all the other satellite campuses that are making use of 256kbps links. It is therefore recommended that these links be upgraded substantially before the deployment of a VoIP system even is considered.
- Secondly, the Internet link, which is currently a 2Mbps link, is inadequate. All the users of the Vaal University network are using this link for Internet access. This link is simply insufficient to satisfy the needs of all the University users and should be upgraded as a matter of urgency.
- If the network on which a VoIP system is going to be implemented contains routers and/or firewalls, a number of ports need to be opened in order for the system to work reliably. If a firewall is used to place calls into the Internet, open the following ports:

UDP (in, out): Ports 5060, 5062, 30000-30021

UDP (in, out): Ports 3478, 3479

TCP (out): Port 80

- If routers are installed in the network, activate "port forwarding" for the following ports: UDP 5060, 5062, 30000-30021. Forward these ports to the computers that you will use to make and receive calls over the Internet or the network that is used for Internet access. This would allow the users of the University network to make use of free Internet telephony services such as Net2Phone, Skype and others which can lead to a vibrant community interacting with each other using voice services, not only domestically but also internationally. This could lead to the sharing of research and subject related information on a virtually real time basis. The creator of a specific product or the inventor of a specific idea would virtually become a staff member in the sense that researchers or students would be able to interact with such a person or organisation on a one to one basis, without the need for expensive international telephone calls, or delays in waiting for e-mail responses.

6.4 Fields for future study

With the recent developments in the transmission of voice over IP networks, the next natural step would be the transmission of video and other multimedia content using existing IP networks. This can have the effect of users on a particular network attending a presentation consisting of all the media types, which are voice, video and data on a real time basis. The research, development and implementation of the glass campus, which is the provisioning of fibre optic links to the users desktop, will make gigabit bandwidth available at the users desktop or living space in terms of student residences. This will allow such users to become active participants at an international conference or study group without having to physically attend it, leading to a dynamic vibrant exchange of knowledge and information.

References

List of sources consulted.

- ARORA, R. 1999. *Voice over IP: Protocols and Standards*: [Online]. Available at: http://www.cse.ohio-state.edu/~jain/cis788-99/VoIP_protocols/index.html. Accessed: 24/5/2004
- CHAPPELL, L.A., TITTEL, E. 2002. *Guide to TCP/IP*: Boston: Course Technology
- DOERING, D. SIMPSON, T. 2000. *Netware 5.0: Network Administrator*: Cambridge Massachusetts. Course Technology.
- FONG, P.J., KNIPP, E., GRAY, D., HARRIS, S.M. et al. 2002. *Configuring Cisco Voice Over IP*, Second Edition: [Online] Available at: http://www.syngress.com/book_catalog/228_VOIP2E/sample.pdf. Accessed 30/6/2004
- FURTADO, J. 2003. *Understanding SIP, Today's Hottest Communications Protocol Comes of Age*: [Online]. Available at: <http://www.ubiquity.net/pdf/Understa.pdf>. Accessed 5/7/2004
- HEYWOOD, D. JERNEY, J.J., JOHNSTON, J. 1992. *LAN Connectivity*: Carmel: New Riders Publishing.
- KUTZ, K.W. 2002. *Extreme Network +*: New York. Element K Press LLC.
- NEWPORT NETWORKS. 2004. *VoIP-MoIP tech note*: [Online]. Available at: <http://www.newport-networks.com/downloads/voip-bandwidth-WP.pdf>. Accessed 26/9/2004
- RADCOM. 2004. *Understanding SIP Servers*: [Online]. Available at:

<[HTTP://www.sipcenter.com/sip.nsf/html/WEBB5YMUHL/\\$FILE/RADVISION_Understanding_SIP_Servers.pdf](http://www.sipcenter.com/sip.nsf/html/WEBB5YMUHL/$FILE/RADVISION_Understanding_SIP_Servers.pdf)>. Accessed 16/8/2004

- SCHULZRINNE, H., ROSENBERG, J. 1999. *The IETF Internet Telephony Architecture and Protocols*: [Online]. Available at:

<[HTTP://www.cs.ucdavis.edu/~prasant/ECS289/ipt1.pdf](http://www.cs.ucdavis.edu/~prasant/ECS289/ipt1.pdf)>. Accessed: 24/6/2004

- SCHULZRINNE, H., ROSENBERG, J. 1998. *Internet Telephony: Architecture and Protocols an IETF Perspective*: [Online]. Available at:

<[HTTP://www.cs.columbia.edu/~hgs/papers/Schu9902_Internet.pdf](http://www.cs.columbia.edu/~hgs/papers/Schu9902_Internet.pdf)>.

Accessed: 24/6/2004

- SCHULZRINNE, H., ROSENBERG, J. 1998. *A comparison of SIP and H.323 for Internet Telephony*: [Online]. Available at:

<[HTTP://www.cs.columbia.edu/~hgs/papers/Schu9807Comparison.pdf](http://www.cs.columbia.edu/~hgs/papers/Schu9807Comparison.pdf)>.

Accessed: 4/6/2004

- SINGH, K. JIANG, W. LENNOX, J. NARAYANAN, S. SCHULZRINNE, H. 2002. *CINEMA: Columbia Internet Extensible Multimedia Architecture*: [Online]. Available at:

<[HTTP://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2002/cucs-011-02.pdf](http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2002/cucs-011-02.pdf)> CINEMA >. Accessed 1/7/2004

- TANENBAUM, A.S. 1989. *Computer Networks*: New Jersey: Prentice Hall Inc.

- VÄÄNÄNEN, K. 1999. *H.323 in Telecommunications*: [Online]. Available at:

<[HTTP://keskus.hut.fi/julkaisut/tyot/erikoistyot/VoIP/44380T.pdf](http://keskus.hut.fi/julkaisut/tyot/erikoistyot/VoIP/44380T.pdf)>. Accessed: 5/2/2004

List of sources quoted.

- CHAPPELL, L.A., TITTEL, E. 2002. *Guide to TCP/IP*: Boston: Course Technology
- FONG, P.J., KNIPP, E., GRAY, D., HARRIS, S.M. et al. 2002. *Configuring Cisco Voice Over IP*, Second Edition: [Online] Available at:
<[HTTP://www.syngress.com/book_catalog/228_VOIP2E/sample.pdf](http://www.syngress.com/book_catalog/228_VOIP2E/sample.pdf)>. Accessed 30/6/2004
- FINGAL, F. GUSTAVSSON, P.1999. *A SIP of IP-telephony*: [ONLINE] Available at: <[HTTP://voip.candra.biz/paper/H323-SIP/Fing9902_SIP.pdf](http://voip.candra.biz/paper/H323-SIP/Fing9902_SIP.pdf)>. Accessed 24/6/2004
- NEWPORT NETWORKS. 2004. *VoIP-MoIP tech note*: [Online]. Available at: <[HTTP://www.newport-networks.com/downloads/voip-bandwidth-WP.pdf](http://www.newport-networks.com/downloads/voip-bandwidth-WP.pdf)>. Accessed 26/9/2004
- PROTOCOLS.COM. S.a. TCP/IP reference page: [Online]. Available at: <[HTTP://www.protocols.com/pbook/tcpip1.htm#MAP](http://www.protocols.com/pbook/tcpip1.htm#MAP)>. Accessed 5/4/2004
- PROTOCOLS.COM. S.a. H.323 Architecture: [Online]. Available at: <[HTTP://www.protocols.com/pbook/tcpip1.htm#MAP](http://www.protocols.com/pbook/tcpip1.htm#MAP)>. Accessed 5/4/2004
- PROTOCOLS.COM. S.a. SIP Architecture: [Online]. Available at: <[HTTP://www.protocols.com/pbook/tcpip1.htm#MAP](http://www.protocols.com/pbook/tcpip1.htm#MAP)>. Accessed 5/4/2004
- RADCOM. 2003. *Fine-tuning Voice over Packet services*: [Online]. Available at: <[HTTP://www.protocols.com/pbook/pdf/VolP.pdf](http://www.protocols.com/pbook/pdf/VolP.pdf)>. Accessed 10/5/2004
- RADCOM. 2004. *Understanding SIP Servers*: [Online]. Available at: <[HTTP://www.sipcenter.com/sip.nsf/html/WEBB5YMUHL/\\$FILE/RADVISION_Understanding_SIP_Servers.pdf](http://www.sipcenter.com/sip.nsf/html/WEBB5YMUHL/$FILE/RADVISION_Understanding_SIP_Servers.pdf)>. Accessed 16/8/2004

- SCHULZRINNE, H., ROSENBERG, J. 1999. *The IETF Internet Telephony Architecture and Protocols*: [Online]. Available at:
<[HTTP://www.cs.ucdavis.edu/~prasant/ECS289/ipt1.pdf](http://www.cs.ucdavis.edu/~prasant/ECS289/ipt1.pdf)>. Accessed: 24/6/2004
- SCHULZRINNE, H., ROSENBERG, J. 1998. *Internet Telephony: Architecture and Protocols an IETF Perspective*: [Online]. Available at:
<[HTTP://www.cs.columbia.edu/~hgs/papers/Schu9902_Internet.pdf](http://www.cs.columbia.edu/~hgs/papers/Schu9902_Internet.pdf)>. Accessed 4/6/2004
- SCHULZRINNE, H., ROSENBERG, J. 1998. *A comparison of SIP and H.323 for Internet Telephony*: [Online]. Available at:
<[HTTP://www.cs.columbia.edu/~hgs/papers/Schu9807Comparison.pdf](http://www.cs.columbia.edu/~hgs/papers/Schu9807Comparison.pdf)>.
Accessed: 4/6/2004
- SINGH, K. JIANG, W. LENNOX, J. NARAYANAN, S. SCHULZRINNE, H. 2002. *CINEMA: Columbia Internet Extensible Multimedia Architecture*: [Online]. Available at:
<[HTTP://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2002/cucs-011-02.pdf](http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2002/cucs-011-02.pdf)> CINEMA >. Accessed 1/7/2004
- TANENBAUM, A.S. 1989. *Computer Networks*: New Jersey: Prentice Hall Inc.
- VÄÄNÄNEN, K. 1999. *H.323 in Telecommunications*: [Online]. Available at:
<[HTTP://keskus.hut.fi/julkaisut/tyot/erikoistyot/VoIP/44380T.pdf](http://keskus.hut.fi/julkaisut/tyot/erikoistyot/VoIP/44380T.pdf)>. Accessed: 5/2/2004

Annexure

Annexure 1	SIP server dial plan	81
Annexure 2	SIP server call log	82
Annexure 3	SIP server specification	83
Annexure 4	Gateway specification	85

Annexure 1 SIP server dial plan

Matching Patterns:

\$request=^INVITE

to=sip:(0.{9})@

Deploy Patterns:

to=sip:%1@192.168.0.4

Annexure 2 SIP server call log

26, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.67.31:5060, 00:03:12, Tue Oct 12 10:26:04 CAT 2004, Tue Oct 12 10:26:33 CAT 2004, Tue Oct 12 10:29:46 CAT 2004, Success, -1

31, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.67.31:5060, 00:00:42, Tue Oct 12 10:29:50 CAT 2004, Tue Oct 12 10:29:52 CAT 2004, Tue Oct 12 10:30:34 CAT 2004, Success, -1

33, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.67.31:5060, 00:00:08, Tue Oct 12 10:33:17 CAT 2004, Tue Oct 12 10:33:19 CAT 2004, Tue Oct 12 10:33:28 CAT 2004, Success, -1

36, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.67.31:5060, 00:00:20, Tue Oct 12 10:36:32 CAT 2004, Tue Oct 12 10:36:39 CAT 2004, Tue Oct 12 10:36:59 CAT 2004, Success, -1

37, sip:20040997lourens@196.21.67.31:5060, sip:Hennie@196.21.67.30:5060, 00:00:09, Tue Oct 12 10:38:54 CAT 2004, Tue Oct 12 10:38:55 CAT 2004, Tue Oct 12 10:39:05 CAT 2004, Success, -1

38, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.67.31:5060, 00:00:08, Tue Oct 12 10:41:11 CAT 2004, Tue Oct 12 10:41:16 CAT 2004, Tue Oct 12 10:41:25 CAT 2004, Success, -1

39, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.67.31:5060, 00:00:17, Tue Oct 12 10:41:30 CAT 2004, Tue Oct 12 10:41:33 CAT 2004, Tue Oct 12 10:41:50 CAT 2004, Success, -1

40, sip:57521vanderschyff@196.21.67.30:5060, sip:Dewald@196.21.65.31, 00:00:00, Tue Oct 12 10:43:44 CAT 2004, Tue Oct 12 10:45:20 CAT 2004, Time Out, -1

Annexure 3 SIP server specification

OnDO SIP Server

Brakeke

datasheet

OUTLINE

The OnDO SIP Server is a call control server with IETF Standard Protocol SIP. The server has the functions of a Registrar, a Proxy Server, and a Call Router. For the local network that OnDO SIP Server locates, it can provide NAT traversal service (NAT). OnDO SIP Server is designed to work with all sorts of SIP phones on the market, it doesn't limit your choices by manufacturer nor vendors. IP telephony system will not only simplify your office space by combining your computer and phone on one cable, but will also provide you the flexibility to expand and grow your business.

FEATURES

- Supports various operating systems -- on Java (JDK version 1.4 or later) environment
- Offers easy administration through any web browser, thus remote administration is feasible
- Able to make revisions even there are active sessions
- Provide wide range of SIP client choices
- Easy to design and set phone-number rules depending on your organization structure by using its flexible Dial Plan function. (Example 1: prefix 1 for Tokyo branch and prefix 2 for US branch. Example 2: Tree phone number structure. Prefix 1 for US branches. Prefix 11 for New York. Prefix 15 for San Francisco.)

FUNCTIONS

• Registrar Service

OnDO SIP Server will receive REGISTER requests from either a client application or UA, and update its database appropriately. Using the registrar function, you will be able to answer calls with any client through your unique SIP-URI.

• Routing

OnDO SIP Server will route SIP requests through a client or other servers to the most appropriate SIP-URI address based on its register database. When a user cannot be located in the database, the Dial Plan setting will be used. If the routing resolves successfully at the proxy server, a caller can establish a call even when the final SIP-URI address is unknown to the call recipient.

• NAT Service

When caller and call recipient are located on different networks, OnDO SIP Server can connect calls by rerouting SIP packet to the appropriate local server or recipient. It is common to have private, local IP addresses within a LAN environment. Thus network address translation (NAT) services are necessary when a local user is establishing a connection with another user with a global IP address. In addition, using NAT services helps protect the security of your local network.

• Dial Plan

You can make flexible routing rules by defining matching rules and filtering rules for headers and IP addresses in the SIP packets using regular expressions in this Dial Plan.

• Session Maintenance

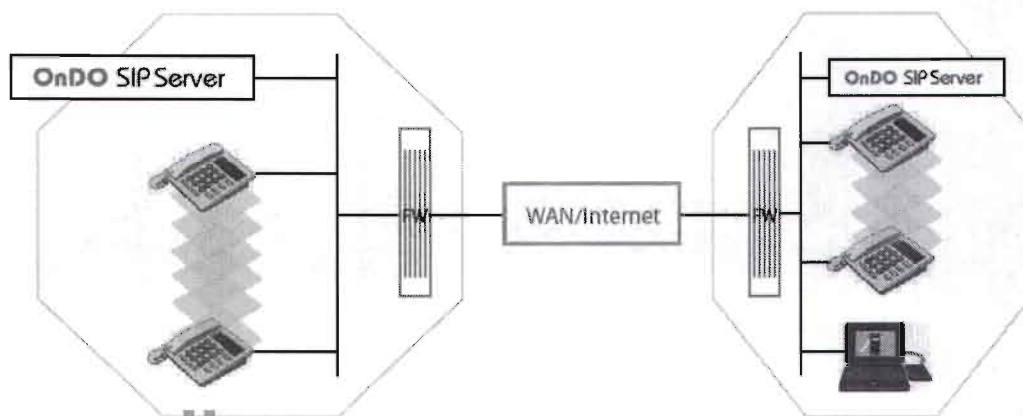
You can check the status or terminate the active calls.

• Logging

Logs for previous sessions can be viewed through a web browser.



SAMPLE NETWORK STRUCTURE



SPECIFICATIONS

♦ VoIP Protocol for signaling	SIP (RFC3261 Standard)
♦ VoIP Protocols for delivery of voice	RTP, RTCP (When using NAT Traversal)
♦ Routing Methods	Register database or Dial Plan
♦ NAT Traversal	Proprietary method
♦ Maximum Number of Simultaneous Connections	50 Connections (Using a Pentium-4 3.2GHz with 512 MB of RAM in a 100M Ethernet network. Connection numbers may vary depending on network condition. Not using a NAT traversal, will increase the number of possible connections.)
♦ Administration	Web-based

OPERATING ENVIRONMENT

♦ Operating System Supported	Microsoft Windows XP/2000, Red Hat Linux v.8.x/9.x
♦ Java™	JDK 1.4 or later
♦ RAM	128 MB minimum

*You need to install Apache Tomcat 4.1.2 or later if you use an OS other than Windows OS.

CONTACT INFORMATION

Brekeke Software, Inc.

4 West 4th Avenue, Suite 404, San Mateo, CA 94402-1614 U.S.A.

TEL: +1 650-401-6636 FAX: +1 650-401-6629

URL: <http://www.brekeke.com/>

email: sales@brekeke.com

Annexure 4 Gateway specification



MGCP / SIP gateway
Advanced 2 ports IP Telephony Gateway

VoIP Analogue Gateway Series

Key Features

- MGCP 1.0 (RFC 2705) or SIP (RFC 3261) compliant
- Supports 2 simultaneous FAX / Voice calls
- 4 Ethernet switch ports with IP sharing functions
- Optional server which enables small businesses to build up private VoIP network (SP model)
- QoS support guarantees voice bandwidth in a busy network
- Supports IP TOS (Type Of Service)
- Internet gateway functions
- T.30 (G III) / Real Time T.38 / Secured T.38 Fax Relay
- Feasible for Fixed IP or dynamic IP network (PPPoE / DHCP client, support DDNS)
- Configurable Hot Line feature
- Supports IP-to-PSTN / PSTN-to-IP applications
- Life-Line support (IP / power failure over relay)
- NAT traversal - STUN and UPnP
- Pass through NAT
- Call Detailed Records (CDR)
- Web-based firmware upgrade
- Caller ID Delivery
 - FXS : DTMF & FSK Call ID generation
 - FXO : DTMF & FSK Call ID detection
- Easy Configuration by IVR and Web-based GUI
- Greeting message

Overview

The 2 ports stand-alone VoIP Gateway carries both voice and facsimile over the IP network. It supports either MGCP or SIP industry standard call control protocols. As a standard user agent, it is compatible to all well known Soft Switches, SIP proxy servers / Call agents (MGCP). While running the optional server software, the gateway can be configured to establish a private VoIP network over the Internet without a 3rd party SIP Proxy Server.

There are three models available: 2 ports FXS, 2 ports FXO or 1 port FXS plus 1 port FXO for different applications. The gateway can be seamlessly integrated to existing network by connecting to a phone set, PBX, key telephone system, fax machine or PSTN line. With only a broadband connection such as ADSL bridge/router, Cable Modem or leased line router, it allows you to gain access to voice and fax services over IP in order to reduce the cost of international and long distance calls.

In addition, the in-built 4 ports switch supports comprehensive Internet gateway functions to accommodate other PCs or IP devices to share the same broadband stream. QoS function allows voice and data traffic to flow through where voice traffic is transmitted in the highest priority. With TOS bit enabled, it guarantees voice packets to have first priority to pass through a TOS enabled router.

With the support of DDNS, it makes the gateway reachable by its domain name where the IP address is dynamically assigned by the ISP. It helps users to host a web site or mail server in a PPPoE or DHCP network. By enabling the CDR function & setting up a simple server, administrators are allowed to log and view all call records such as call duration, time and date of calls and latency, etc.



OCTTEL COMMUNICATION CO., LTD
www.octtel.com.tw



Technical Specifications

Telephony Specifications

- Voice Algorithms: G.711(A / μ law), G.726, G.729A and G.723.1
- Fax Support: T.30 G III, Real Time / Secured T.38 Fax relay
- Silence Suppression: G.711, G.726, G.729A, G.723.1
- Echo Cancellation: G.165/G.168 compliant
- Voice Activity Detection (VAD) & Comfort Noise Generation (CNG)
- Dialing: DTMF, PULSE (optional)
- Signaling Protocol: Loop Start
- Adaptive Jitter Buffer and Programmable Gain Control
- Phone/ Fax/ Key Phone System/ PBX Interface: FXS analogue (RJ-11)
- PSTN Interface: FXO analogue (RJ-11)
- Polarity Reversal:
 - *FXS generation
 - *FXO detection

IP Specifications

- IETF MGCP v1.0 (RFC 2705), SIP (RFC 3261) compliant
- LAN -
 - * Interface : 4 Ports Ethernet switch, 10/100 Base-T, RJ-45
 - * Internet Gateway Functions : DHCP server, NAT, Virtual Server, DMZ, IP/PORT/MAC filtering
- WAN -
 - * Interface : 10/100 Base-T Ethernet, RJ-45
 - * PPPoE client, DHCP client, Fixed IP address
 - * NAT Traversal : STUN, UPnP
 - * Dynamic DNS (DDNS)
- QoS and IP TOS

Call Features

- Call hold
- Call waiting
- Call forward
 - Unconditional (follow me)
 - Busy forward
 - No answer forward

- Call transfer
 - Unattended transfer
 - Attended transfer
- Call pick up
- Repeat dialing

Configuration & Management

- Web-based Graphical User Interface (GUI)
- Interactive Voice Response (IVR)
- Remote management over the IP network
- Web-based firmware upgrade

LED Indicators

- Power, Run, Alarm, WAN, FXS/FXO (P1, P2) and LAN (L1~L4)

General Specifications

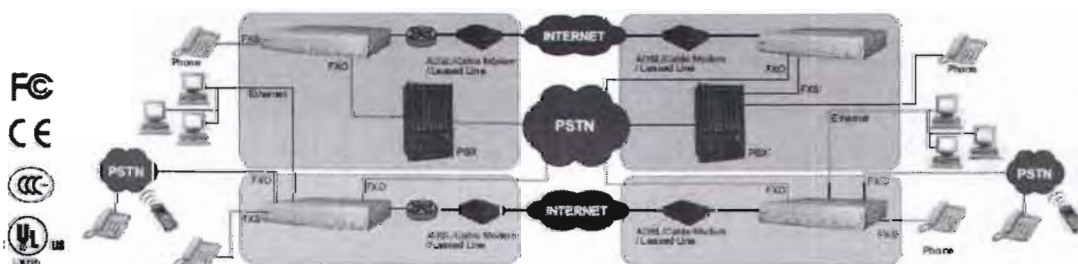
- AC Power: AC100V- 240V, DC12V/1.2A, 50/60 Hz
- Temperature: 0°C to 45°C (Operation)
-25°C to 75°C (Storage)
- Humidity: up to 90% non-condensing
- Emissions: FCC Part 15 Class B, CE Mark, CCC
- Telephone Safety: FCC Part 68
- Dimension: 202 x 172 x 35 mm (W/D/H)
- Weight: 430g

Ordering Information

Model	Description
MG4220DXO	1FXS, 1FXO, 1WAN, 4LAN, MGCP
MG4220DXS	2FXS, 1WAN, 4LAN, MGCP
MG4220DX2O	2FXO, 1WAN, 4LAN, MGCP
SP4220DXO	1FXS, 1FXO, 1WAN, 4LAN, SIP
SP4220DXS	2FXS, 1WAN, 4LAN, SIP
SP4220DX2O	2FXO, 1WAN, 4LAN, SIP

The specifications are subject to change without notice. All other products or trademarks are the property of their respective owners.

Applications



Octtel OCTTEL COMMUNICATION CO., LTD
 Headquarter : 7F-1, No.300, Daduan 10th St., Taichung, Taiwan 4018 R.O.C.
 TEL: 886-4-2252-0199 FAX: 886-4-2252-3340
 Taipei Marketing Center:
 7F, No. 20, Lane 478, Ruelgong Rd., Neihu District, Taipei, Taiwan, R.O.C.
 TEL: 886-2-6751-8200 FAX: 886-2-6751-8138
 e-mail: sales@octtel.com.tw http://www.octtel.com.tw

Distributor / Retailer :