150013

SIMULATION ON EFFICIENCY OF IP PROTOCOL IN NETWORKS

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

BY

BARBAROS PREVEZE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

JUNE 2004

Approval of the graduate school of natural and applied sciences.

Prof. Dr. Yurdahan GÜLER
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. R. Mehmet TOLUN
Chairman of the Department

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

Asst. Prof. Dr. Halil T. EYYUBOĞLU
Supervisor

Examining Committee Members

Prof. Dr. Turhan ALPER

Assoc. Prof. Mustafa ALKAN

Asst. Prof. Halil T. EYYUBOĞLU

ABSTRACT

Simulation on efficiency of IP protocol in networks

Preveze, Barbaros

Ms, Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Halil T. Eyyuboğlu

June 2004, 53 pages

This thesis presents the results of a computer simulation for delays experienced by voice over internet protocol (VoIP) packets riding over an enterprise packet network connected to public internet. The results indicate that among these delays line speed, MTU (maximum transferable unit) and packet size play the most decisive roles.We also have proven that ,IP Phone is much more preferable to using PC's in providing the expected quality for voice transmissions over the internet protocol.

Key words: VoIP, Delay Factors, QoS ,Jitter

ÖZ

IP protokolünün network üzerindeki verimliliğine ait simülasyon

Preveze, Barbaros

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Yöneticisi: Yrd Doç. Dr. Halil T. Eyyuboğlu

Haziran 2004, 53 sayfa

Bu tez internete çıkışı olan bir ağın internet protokolü üzerinden gönderdiği ses paketlerinin etkilendiği ağ parametrelerini ve uğradığı gecikme miktarlarını saptayan bir simulasyon programının sonuçlarını sunmaktadır. Elde ettiğimiz sonuçlar bize ağ parametrelerinin en etkililerinin hat hızı, maksimum gönderilebilir paket büyüklüğü ve ortalama paket büyüklüğü olduğunu göstermiştir.Tezde Ayrıca beklenen ses kalitesini sağlamak için IP Telefonlarının, PC kullanımına gore çok daha fazla tercih edilebilir olduğu ispatlanmıştır.

Anahtar Kelimeler: VoIP, Gecikme Faktörleri, Servis Kalitesi ,Delay değişkenliği

To my family; Behiye, Nazım and Nazan Preveze

ACKNOWLEDGEMENTS

I express sincere appreciation to Asst. Prof Halil T. Eyyuboğlu for his guidance and insight throughout the research. Then I must thank Assoc. Prof Yahya Baykal for his encouragements. My friends Çağlar Arpalı and Serap Altay are gratefully acknowledged. To my parents, I offer sincere thanks for their unshakable faith in me and their willingness to endure with me the vicissitudes of my endeavors.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

ABBREVIATIONS

ADC	analog to digital converter
ADPCM	Adaptive differential pulse code modulation
APL	average packet length
ATM	asynchronous transfer mode
CODEC	coder – decoder
DIFFSERV	differential services
DQD	data queuing delay
ETSI	European telecommunication standards institute
FEC	forward error correction
IP	internet protocol
IPv4	internet protocol version 4
IPv6	internet protocol version 6
ITSP	internet telephony service provider
ITU	international telecommunication union
PC	Personal computer
PCM	pulse code modulation
PD	propagation delay
PSN	packet switching networks
PSTN	public switching telephone network
QoS	quality of service
RSVP	reservation protocol
RTP	real time transfer protocol
SD	serialization delay
SLA	service level agreement
TCP	Transport control protocol
TOS	type of service
TTL	time to live
UDP	user datagram protocol
VDCD	voice data contention delay
VOCODER	voice coder

VoIP	voice over internet protocol
VOPN	voice over packet network
VVCD	voice / voice contention delay

CHAPTER 1

INTRODUCTION

Telecommunication has a great importance in business life. It is known that since the most widely used data type is voice, the associated cost becomes an amount not to be neglected. This causes some people to prefer VoIP instead of using normal PSTN. In VoIP if it is streaming voice application then it can simply be treated of as ordinary data thought as ordinary data , but if it's a real time voice application, then we will face a problem of providing the right performance and quality for the users. This is already offered by Public Switched Telephone Networks (PSTN) but since a permanent capacity is allocated for each conversation and since these networks have been designed to operate at channel rates, which are in multiples of 64 kbps, the network bandwidth will be wasted in slower rated flows and consequently resulting in high costs for lengthy conversations. So an alternative solution, which is VoPN (Voice over Packet Networks) is proposed. In PSN (packet switched networks) the data and voice are integrated in a single network and they both use all available capacity on the network at the same time. But since some delays occur for each packet traversing the links and since it is also possible for the packets to follow a different path through the destination the jitter problem (variability of delay) occurs for the real time applications.

Packet switching was invented allowing voice streams to be fragmented into packets where each one could be routed along different paths. For this type of transport mechanism, varying degrees of voice compression was also foreseen. Packet switching is facilitated by techniques such as IP, ATM and frame relay. But in each of these, due to processing of packets, delay becomes the major parameter that must be monitored closely. This is particularly so for delay sensitive applications such as voice transmission. The forerunner of the packet switching technologies, IP, was developed with only computer to computer communication in mind. Hence no prime consideration had been given to packet delays.

The others such as ATM attempted to lessen these delays by selecting optimum packet parameters. A detailed comparison of such technologies is given in an earlier work [1] , and will not be repeated here. Now we have the situation that, transmission of voice packets over IP networks, named as VoIP, has become more popular and more future promising than all other alternatives. In this thesis factors effecting the delay are explored and a simulation program is written using Matlab and we have illustrated. The delay graphs generated are illustrating the variation of delays by each delay factor taking into account of ETSI delay recommendations. The text is based on an investigation of delays for VoIP packets in a given network configuration and is organized as follows. Chapter 2 explains the general overview of VoIP, in Chapter 3, quality of service will briefly be explained , and overall concepts of latency, jitter and their general impact are explained. In Chapter 4, the factors affecting the total delay are examined. The implementation structure of our simulation program is contained in Chapter 5. Next we present and discuss the results of our simulation program in Chapter 6. We compared IP Phone vs. PC implementation for the delay amounts in Chapter 7, Finally we end with conclusions in Chapter 8.

CHAPTER 2

VoIP OVERVIEW

Voice over internet protocol (IP) network is a general term that refers to any means of converting voice calls into voice data packets that are transmitted over an IP network which is a packet-oriented network, designed to transport packets of data between systems. Systems are assigned IP addresses that identify them to the IP network. This network is not designed for the reliable transport of packets between systems, i.e., it does not guarantee packets transmitted by one system will safely arrive at the intended recipient system. It is left to higher-level protocols such as TCP to guarantee correct and reliable delivery. IP packets are not related as far as the IP network is concerned.

2.1 Voice over IP vs. PSTN

When a PSTN line is used, typically payment is made according to time of occupation. This means the more time the line is used the more payment will accumulate, Furthermore the bandwidth is always in multiples of 64 kbps i.e. it can not be sliced. In addition it is almost impossible to hold another simultaneous session. With VoIP mechanism on the other hand, the cost is nearly independent of time duration and multiple sessions may be set up.

2.2 Codecs

Vocoder is an acronym for voice coder/decoder. This is the term given to the process of encoding and decoding voice using an algorithm implemented in software and/or hardware. This function is usually performed by hardware or DSP software. In this process voice is compressed and decompressed. Common vocoder algorithms are G.711 (PCM), G.723.1 G.726, G.728 and G.729. In this text the term codec will mean of voice coder/decoder. A G.729A type vocoder is employed in our simulation program. This vocoder, sometimes known by the abbreviation CS-ACELP (Conjugate Structure Algebraic-Code Excited Linear Prediction), is deployed in many VoIP gateways. It has a bit rate of 8 kbits /second. The related vocoder algorithm has silence suppression, meaning silence is not transmitted and does not occupy bandwidth. Table 2.1 illustrates, coding / encoding rates and best/worst case delay amounts for different codec types.

Table 2.1: Best and worst case processing delay

Coder	Rate	Required Sample block	Best Case Coder Delay	Worst Case Coder Delay	length
ADPCM, G.726	32 Kbps	10 ms	2.5 ms	10 ms	20 bytes
CS-ACELP, G.729A	8.0 Kbps	10 ms	2.5 ms	10 ms	10 bytes
MP-ACELP, G.723.1	5.3 Kbps	30 ms	5 ms	20 ms	20 ms

Since calculations are made according to worst case conditions, the delay amount is taken as 10 ms for G.729A codec type in the simulation program.

2.3 Overview on a VoIP connection

To setup a VoIP communication, first we need the ADC to convert analog voice to digital signals (bits) then the bits have to be compressed in some format for transmission, Here we have to insert our voice packets in data packets using a real-time protocol such as RTP over UDP over IP (see Figure 2.1)

We need a signaling protocol to call users, such as ITU-T H323.

At receiver side, packets are disassembled, data is extracted, then converted to analog voice signals and finally coupled to a telephone equipment (or sound card if a pc is used)

All that must be in real time.

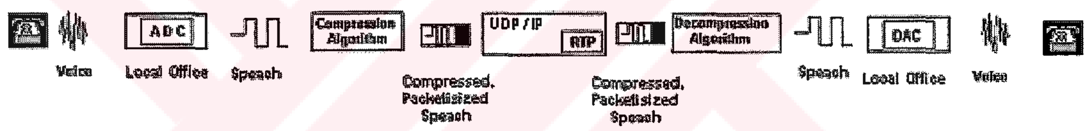


Figure 2.1 : Base architecture of VoIP

2.5 A typical telephone call process of VoIP

The general flow of a two-party voice call using voice over IP is as follows:

1. When user picks the handset up, this will send off-hook condition signals to the signaling application part of Voice over IP.

2. Then the session application part of Voice over IP issues a dial tone and waits for the telephone number to be dialed.

3. The user dials the telephone number and the session application accumulates and stores those numbers.

4. After accumulation of enough digits in order to match a configured destination pattern, the telephone number is mapped to an IP host via the dial plan mapper. The IP host has a direct connection to either the destination telephone number or a PBX that is responsible for completing the call to the configured destination pattern.

5. The session application then runs the H.323 session protocol to establish a transmission and a reception channel for each direction over the IP network. If RSVP has been configured, the RSVP reservations are put into effect to achieve the desired quality of service over the IP network.

6. The codecs are enabled for both ends of the connection and the conversation proceeds using RTP/UDP/IP as the protocol stack.

7. Any signals that can be carried in-band are cut through the voice path as soon as end-to-end audio channel is established

8. When either end of the call tears down, the RSVP reservations torn down (if RSVP is used) and the session ends. Each end becomes idle, waiting for the next off-hook condition to trigger another call setup.

2.6 Analog to digital conversion

This is made by hardware, typically by card integrated ADC.

Today sound cards allow coding in 16 bits that will turn a 22 kHz analogue voice into 176 kbps.

But for VoIP, we do not need such a high throughput (176 Kbytes/s) to send voice packet.

2.7 Compression algorithms

It is known that classic voice bandwidth is 4 kHz, so sampling rate has to be 8 kHz (due to Nyquist Theorem).

If we represent each sample with 8 bit (for 256 possible values).

Throughput is 8000 Hz *8 bit = 64 kbps, the rate of a typical digital phone line.

CS-ACELP, (Standard ITU-T G.729 and G.729a) reduces this to 8 kbps.

2.8 RTP / UDP / IP

Voice over IP involves more than just IP. Immediately above IP in the protocol stack is UDP, and above that is RTP. The Functions Performed by these protocol layers are summarized in Figure 2.2 The header structure is shown with 4 bytes (32 bits) horizontally, following the convention which is useful for 32 bit implementations. Here those features of RTP /UDP /IP that are relevant to Voice [2] are described.

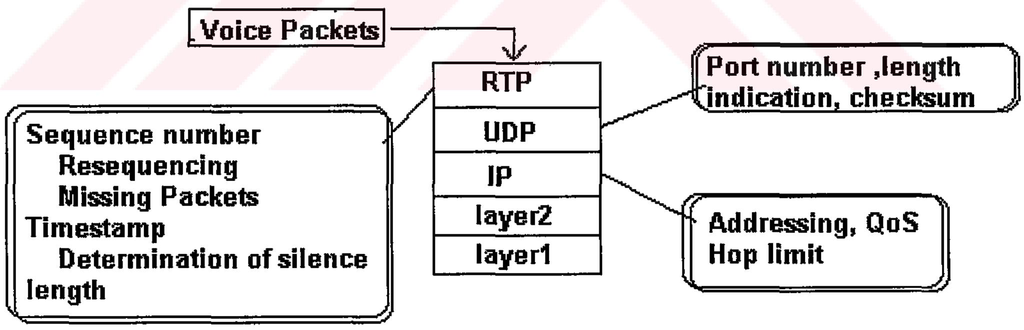


Figure 2.2 : Transporting a voice packet through RTP / UDP / IP structure

The RTP header is 12 bytes providing:

Sequence number :

The sequence number is used to re-sequence the miss-sequenced packets and the lost packets can also be determined from the gaps using the sequence number.

Timestamp :

There will be no voice packets generated during the silence and the destination will detect these intervals using timestamp and it will generate silence or background noise.

Synchronizing source ID :

In a multiparty conference the synchronizing source ID identifies which participant in the conference is speaking. RTP Header structure is illustrated in figure 2.3.

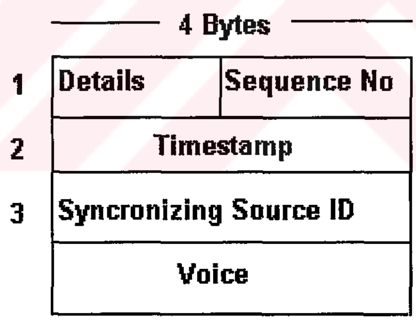


Figure 2.3: RTP Header

The UDP header is 8 bytes providing:

Source and destination port numbers :

UDP port numbers allow multiple voice calls to share a single IP source or destination address. This is essential for ITSP gateways, which simultaneously receive and send many different calls.

Length :

Voice codecs generate different lengths of voice packets and hence different length of UDP packets. The source sets the length indicator in order to inform the destination about the length of the packet.

Checksum :

If errors are introduced during the transmission they can be recalculated using the checksum.(see Figure 2.4)

	4 Bytes	
1	Source Port #	Destination Port #
2	Length	Checksum
	RTP Packet	

Figure 2.4: UDP Header

The IP header is 20 bytes and the services it provides are :

Type of service (ToS) :

It has 1 byte of length and it can be used to assign different QoS to voice traffic.

Time to live (TTL):

This number decreases each time the packets traverses a network node if it becomes zero it will be discarded by the last node, so that the misrouted packets will not circulate in the network forever.

Source / Destination address :

Addresses provide unique addresses for the source and destination of the packet across the internet.

One byte can be used for assigning different QoS to voice traffic compared to other traffic as described.

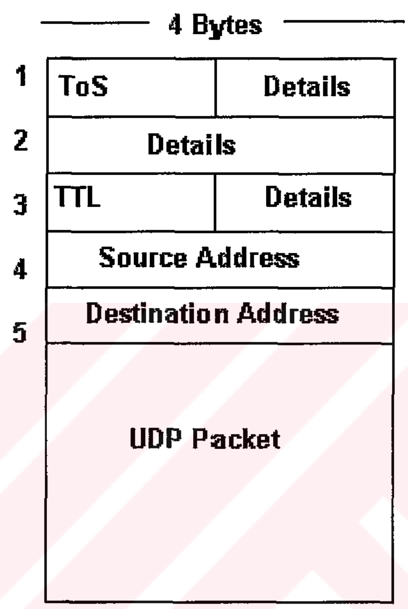


Figure 2.5 : IP Header

2.9 Frame packing

After packetisizing the voice frames into packets, if we make a simple calculation, we will see that the overhead rate of the packet will be 40/50 i.e. 80 % for IPv4 and 52/62 i.e. % 84 for IPv6 (see Table 2.2) [2].

By frame packing, we can create two advantages by packing two voice packets into a packet.

First, the percentage of overhead is reduced by sampling n packets, instead of a single packet, hence the overhead percentage will become (52/(52+10n)) for IPv6 and (40/(40+10n)) for IPv4.

Second, total number of packets traversing the routers will be reduced by 1/n, this will reduce the load on the network routers. But in real time voice applications, this method is unfortunately not applicable, because the voice samples packed into the packet will wait for other packets in order to be packed into the same packet and this will create 10n ms additional delay. On the other hand the delay at the destination side will also increase .

Table 2.2 : IPv4 and IPv6 header sizes

	IPv4	IPv6
IP	20	32
UDP	8	8
RTP	12	12
Voice	10	10
TOTAL	50	62

2.10 IP QoS

The QoS can be provided by two ways in IP. These are :

1. Reservation Protocol (Integrated Services)
2. Diffserv (Differentiated Services)

2.10.1 Reservation protocol (RSVP)

RSVP involves a call setup process in which the user requests QoS and bandwidth, and the routers state the available resource information then the call can be established if and only if the network and the user accept the network conditions.

When the source starts sending packets, they are monitored for ensuring that the source does not exceed the agreed upon traffic specification. In this protocol each packet has its own flow identifier and each router that the packet traverses analyses this field in order to decide whether or not this packet has a bandwidth reservation (using RSVP) or not.

RSVP has 2 modes which are ;

a. Controlled Load

The user requests only bandwidth but not QoS so there is no quantitative guarantee of QoS.

b. Guaranteed

The user can request both bandwidth and QoS. But in RSVP, QoS factors such as accuracy of data sending rate can not be specified, however accuracy is not the major QoS factor for real time voice applications.

RSVP is concerned with providing different levels of QoS for different nodes analyzing the flow identifiers, according to the type of information they send being aware of that QoS is very important for real time applications, while it has no importance for an ordinary data transmission.

Once a connection path is established, all packets traverse the same route that means the connectionless internet behaves as connection oriented by RSVP.

2.10.2 Differential services (Diffserv)

Diffserv defines a different level of QoS for each IP Packet traversing the network . The packets record the behaviors of each node that it traversed and keeps them in Diffserv code point (DSCP) which is stored in ToS field. (see Figure 2.5)

The DCSP is used to specify the "per hop behavior" that the packet receives from the network node. This behavior can include accuracy, delay and bandwidth, without any quantitative specification of delay or accuracy [2].

There are three types of Diffserv service classes which are;

a. Expedited (virtual leased line)

This is the best class of service in terms of accuracy, delay and bandwidth. The network becomes a simulation of constant bit rate (CBR) leased line.

b.Assured Forwarding

There are 12 assured forwarding per hop behaviors, with 3 levels of accuracy and 4 levels of delay. In assured forwarding, different levels of priorities can be assigned for each packet by using these accuracy and delay levels.

c.Class Selector

There are 8 class selector code points. The higher the numerical value of the code point is, the better the per hop behavior from the point of view of delay. The edge router assigns a drop precedence value according to the packet arrival time for each packet, and this value determines the probability of dropping this packet in congested routers. Since the flow rate control is done only at edge routers unlike the RSVP we can say that Diffserv is better than the RSVP in congested networks.

CHAPTER 3

QUALITY OF SERVICE

3.1 Measurement of quality

QoS is what level of service the network is capable of offering and what the user expects from the network

Any kind of service provider has to provide a level of quality for the user. Therefore the vendor and the user sign SLA (service level agreement) for the level of this quality. If the vendor can not provide this quality then they have to pay a penalty to the user. If there are multiple vendors working together, in this case only the responsible one pays the required penalty to the user.

3.2 QoS factors

When the QoS is specified there are some number of factors which have to be taken into account. Which are;

1. Accuracy : Concerns data loss and error control.

2. Latency : Concerns the delay which occurs during the data transmissions.

3. Jitter : Concerns delay variability occurred during the packet transmission.

And the QoS factor only valid for voice packets is

4. Codec quality: Concerns with the voice quality of the codec which makes the conversion from analogue to digital. Effects of codec to the QoS is already explained in section 2.2

There is a very strong relation between accuracy and latency. The acceptable accuracy levels are determined by the acceptable latency levels. Now we investigate these factors.

3.2.1 Accuracy

To determine the accuracy rate, we should know "why do packets get lost? "

Buffer Overflow :

When arrival times of some packets at the output buffer differ, some of the packets may be discarded.

Network Failure:

When some faults on a network component occurs or when a fault in transmission device occurs this causes some stream loses.

3.2.1.1 Recovering the lost packets

The lost packets can be recovered by either sender or receiver. The methods used by the sender are called "sender based methods" and the methods used by the receiver are called " receiver based methods". (see Figure 3.1)

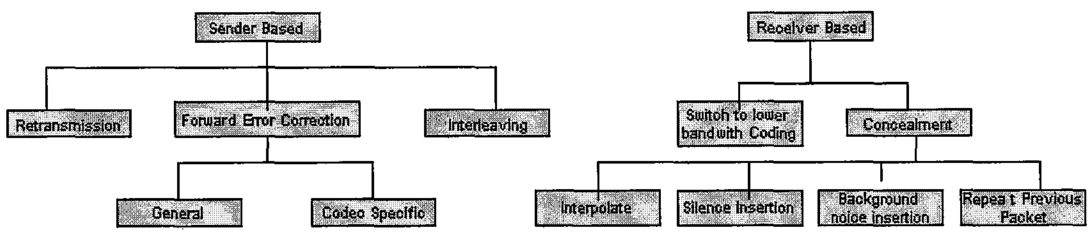


Figure 3.1: Sender based and Receiver based data recovering methods

3.2.1.1.1 Sender based methods

The methods and algorithms applied by sender are called sender based algorithms and these are;

a. Retransmission

Retransmission is the most widely used sender based solution. it works with acknowledgement and since it creates an additional delay it is not preferred in interactive applications. In multi-user systems if two nodes simultaneously request the packet again, the delay between them avoids sending the data twice and this problem is solved by forward error correction.

b. Forward error correction algorithm

In forward error correction method there are two algorithms and these are general purpose algorithm and codec specific algorithm, in general purpose algorithm, there is an additional packet called forward error correction (FEC). Packet which keeps the information about other packets. The sender generates FEC packet and the lost packets can be recovered by combining the information in this packet with the unlost packets as illustrated in Figure 3.2.

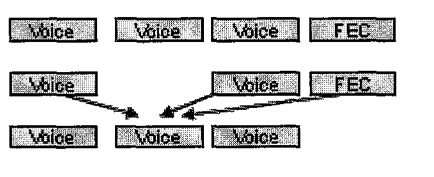


Figure 3.2: Forward error correction general purpose algorithm.

In other forward error correction algorithm called codec specific algorithm, each packet has the compressed version of the previous one and if any of the packets gets lost , the next packet will be able to regenerate the lost packet by decompressing its trailer as shown in Figure 3.3.

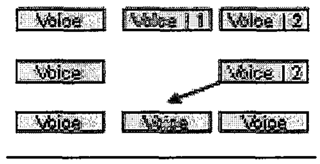


Figure 3.3: Forward error correction codec specific algorithm.

But in congested networks, some group losses may occur and since the consecutive packets will be lost at the same time, the lost packets may not be regenerated. For solving this problem, the interleaving method is employed.

c. Interleaving

In this algorithm all packets are sent in miss-sequenced order and in case of group losses the compressed version of the lost packets will not be in the lost group (see Figure 3.4) . This will give us chance of regenerating the lost packets

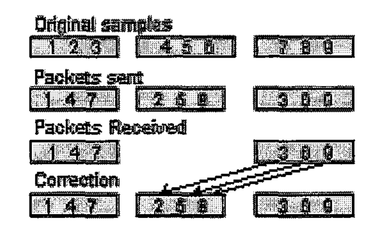


Figure 3.4 : Interleaving algorithm

3.2.1.1.2 Receiver based methods

The methods and algorithms applied by the receiver are called receiver based algorithms. Since less delay occurs in receiver based algorithms, these algorithms are much more preferred than sender based ones for real time applications, but unfortunately they can not find the exact solution of recovering the lost packets. These algorithms are;

a. Switching to Lower Bandwidth :

At the beginning, the users request high bandwidth but in high bandwidth rate the congestion rate and packet loss rate will increase while the voice quality will decrease. In this case the switch temporarily switches to lower bandwidth until the network status returns to the normal conditions.

b. Concealment :

This method hides the packet losses and errors instead of correcting or recovering them. This method is widely used in the cases that the interactivity and the quality is important such as voice transmission. The standard codecs like G.723,G.728 and G.729 use this algorithm.

There are 4 options of hiding the errors and the lost packets.

1. Replacing lost packets with silence

2. Replacing lost packets with background noise

3. Replacing the lost packet with the previous packet. By this way the processing time will decrease to minimum.

4. The last 20 ms is repeated and the voice quality of next 300 ms is decreased.

3.2.2 Latency and Jitter

When a source node sends a data packet to destination, this packet is subjected to several operations during transmission. Each of these operations lasts for a finite time duration. The time that passes for the packet to traverse from source to destination is termed latency (delay). Latency is somewhat insignificant for non-real time applications such as fax, mail etc. But in real time applications such as voice conversations or video conferences, delays have major effects on the quality provided to the user. In voice transmission, a delay exceeding 150 ms creates an echo. When the delay increases above 400 ms, the conversation becomes unintelligible [3,4]. Another cause of delay is the retransmission of packets in case of data losses. But resending of packet is not recommended for real time applications.

There are two types of delays, which are in turn named as fixed and variable [4,5,6]. Fixed delay is attributed to the intrinsic treatment of source signals by the network elements and is not dependent upon loading conditions. Variable delays on the other hand are incurred by the contention of data and voice packets queuing at network nodes. Total delay is calculated by adding fixed and variable delays. According to ETSI standards, this should not exceed 400 ms.

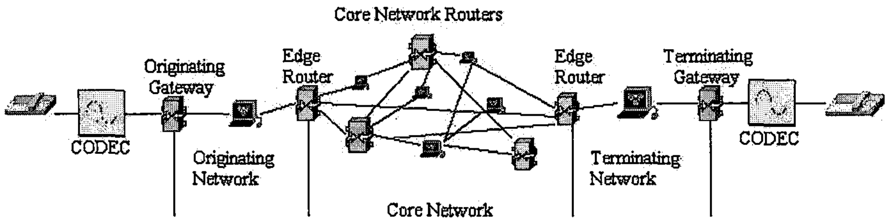
In actual transmissions, packet delays will not be constant but will have a certain statistical distribution. This variation is termed jitter. A buffer, called jitter buffer is used at destination in order to equalize these variations. This way, the variable delay is doubled however [2,3,5,6].

CHAPTER 4

ANALYSIS OF DELAY FACTORS FOR VoIP

IP networks operate on packet switching principles; hence voice in an IP network would be transmitted to the destination as a collection of packets where each one might traverse the network through different routes. If alternative routes were to be composed of different types and numbers of network elements, then each individual packet might arrive at the destination within a different time instance.

In this part of the thesis, the delay parameters involved in our simulation are described. We assume the overall network architecture and the input parameters of the simulation program to be as depicted in Figure 4.1. Here the blocks up to the border of the core network constitute the originating and terminating parts respectively. Typically these would be the in-house portions of an enterprise network. The part lying in-between, probably internet or some public WAN (i.e. shared medium) is termed the core network. In our sense, the packet flow is assumed to be from left to right. For the flows in the opposite direction, the terminology of originating and terminating is interchanged. Unless stated otherwise, originating and termination parts are regarded to be identical in view of delay calculations.



Parameter Name Used in Simulation	Meaning of Parameter Name
Packet Size	Size of Packet in Bytes
Number of Hops in Originating Network	Number of Routers in Originating / Terminating Network
Speed of Core Network	Speed of Core Network
Number of Hops in Core Network	Number of Routers / Switches in Core Network
Speed of Originating Network	Speed of Originating / Terminating Network
MTU	Maximum Transferable Unit
Distance	Distance Between Source and Destination (km)

Figure 4.1 : Network structure used in simulation program

On the originating side, the analog signal generated by the telephone set is digitized into pulse code modulation (PCM) signals by the voice codecs. Then the PCM samples are compressed and converted into packet format, thus ready to be sent across the internet.

For some network configurations, the edge router may also perform codec and compression functions or a PBX installed on enterprise premises may be employed for this purpose. In the subsections below, we attempt to describe various kinds of delays together with their associated formulation in units of milliseconds (ms).

4.1 Look ahead delay

Normally codecs conforming to different standards would use voice blocks of different sizes. For example the block size used by the G.729 A is 10 ms, whereas codecs of G.723.1 have block sizes of 20 ms.

The present packet is checked while compressing the previous one. But at the very beginning of transmission, such an operation is not possible of course. This way, a fixed delay of 5 ms [2] called “Look Ahead Delay” occurs in this section of packet voice transmission.

4.2 Packetization delay

a. Encoding and compression delay:

In VoIP applications, generally the G.729 A type codec is used having a mean opinion score (MOS) value of 4.2 which almost provides toll quality [6]. This type of codec has an 8 kbps modulation rate, hence producing a 10 ms delay for encoding the packet [2, 3, 5, 6, 7, 8].

b. Packetization delay:

The time spent for packetization of encoded and compressed data packets is known as packetization delay. To support a good quality conversation, the packetization delay should be less than 30 ms. [3]. The packetization process commences after storing the packets into a buffer. Here as long as the buffering time remains below 10 ms, the compression and buffering periods will overlap and there will be no additional delays introduced at the buffer [3]. Otherwise, the buffering time would exceed the compression time, and the remaining period would add to the total delay created during the buffering operation.

4.3 Serialization delay (SD)

Serialization of the data/voice packets is going to be implemented at the ingress point of the originating network. And this creates an extra delay, which may be calculated as ;

$$SD = (Voice + 48) \times 8 \,/\, Link\ Speed\ (ms) \tag{4.1}$$

Here the term Voice corresponds to the average number of bytes stored in payload of the packet and 48 indicates the number of bytes added by real time protocol, user datagram protocol and internet protocol (RTP/UDP/IP) headers including some optimizations [2,5].

4.4 Network delays

An IP packet while traversing the whole network suffers several other delays whose sources are mentioned below.

a. Switching Delay

On average, the packets wait for 10 ms at each switch in the originating network and 1 ms at each switch in the core network [5]. Hence the total switching delay becomes

Switching Delay = Number of Routers in Originating Network x 10 + Number of Routers in Core Network x 1 (4.2)

b. Propagation delay (PD)

It's well known that an electronic signal has a velocity, which is at two thirds of the speed of light. So by a simple calculation, we conclude that each km of travel will contribute an extra 5 μs delay. Thus the propagation delay may be written as ;

PD = Distance (in km) x 5 x 0.001 (4.3)

c) Data queuing delay (DQD)

The packets are fed into the network in a serial fashion, so they will line up in a queue waiting to be served sequentially. But depending on the packet size, the waiting time of each packet will be different. Furthermore these queuing delays will also vary depending on IP packet size transmitted. Because of these complications, data queuing delay is not easy to formulate. In our simulation program, we assume that IP is carrying data and voice, but no video packets. In this way voice packets will have the highest priority and may therefore jump ahead of the other packets. Even with this assumption, an accurate prediction of queuing delay is still impossible, but may be simplified to [5];

$$DQD = (APL \times 8000) / (2 \times \text{Link Speed}) \quad \text{(ms)} \tag{4.4}$$

Where APL represents the average packet length and DQD represents the data queuing delay under normal conditions....

d) Voice/Data contention delay (VDCD)

Voice data contention delay is generated as a result of voice and data contention. Under worst case conditions, this delay can be expressed as;

$$VDCD = ((MTU + 48) / \text{Link Speed}) \tag{4.5}$$

In this equation, the term MTU (maximum transferable unit) appears, because in worst case assumption, the maximum payload size of the data packet will be equal to MTU.

e) Voice / Voice contention delay (VVCD)

In an IP network, the voice packets not only contend against the data packets, but they also have to contend with other voice packets for the share of bandwidth. In our case, the serialization delay of the packets, the number of routers that the packet passes through and link usage rates of the voice and data packets all have major effects on the delay created by this contention. The following equation is written to account for all these effects [5] ;

$$VVCD = 2 \times SD \times N \times (\sigma / 2) \times (1 - \sigma) \tag{4.6}$$

Where σ represents link utilisation

Where N denotes the number of routers that the packet has to pass through, Voice denotes total size of voice packets in the network and Data denotes the total size of data packets in the network.

f) Jitter buffer delay

Each packet serialized to the network will face different circumstances of queuing and congestion. So the packet arrivals will not be co-incident. This variability would make the conversation uncomfortable in VoIP systems. Hence a buffer, named as the jitter buffer is used at the receiver side in order to equalize these transmission-oriented fluctuations. But this compensation scheme will cause an additional delay whose magnitude is equal to the maximum jitter size.

E) Voice decoding delay

The encoded packets have to be decoded at the receiver side. The decoding time for each packet is shown to be ;

$$\text{Voice Decoding Delay} = \text{Encoding Delay} \tag{4.7}$$

Since a G.729 A codec imposes an encoding delay of 10 ms ,our decoding delay will simply be 10 ms for each packet [2, 3 5, 6, 7, 8].

CHAPTER 5

SIMULATION PROGRAM

In writing the simulation program, our main objective was to determine the acceptable ranges for each of these delay factors. But since they are closely related to each other, all would have to be taken into consideration simultaneously. In order to observe the specific behavior of a particular delay factor, our simulation program sets all other delay parameters, except the one to be examined, to some default values. Then by varying the numerical value of the delay parameter in question, we are able to evaluate its independent behavior. By repeating this act for each factor, a series of individual graphs covering all factors shall have obtained. The default values, minimum and maximum values (i.e. the observation ranges) used for each parameter and number of samples is listed in Table 5.1. In the simulation program, the step size of an individual factor is adjusted by dividing the difference between the maximum and the minimum values and by the number of samples, i.e.;

$$\text{Step Size} = \frac{\text{Maximum Value} - \text{Minimum Value}}{\text{Number of Samples}} \tag{4.8}$$

In this formulation, number of samples is taken as 100 for all cases.

As mentioned above, our voice packets are assigned the highest priority and video transmission is excluded from our case. Furthermore the preceding formulations already take into account the traffic loading of other existing data sources. Hence it is justifiable that our simulation considers a single voice over IP packet source.

Table 5.1 : The parameter values used in simulation program

Parameter Name	Default Value	Minimum Value	Maximum Value
Packet Length	20 bytes	1 byte	300 bytes
Speed of Core Network	512 kbps	1 kbps	4096 kbps
Speed of Originating /Terminating Network	128 kbps	1 kbps	4096 kbps
Originating/ Terminating Network Hop count	1 hop	1 hop	20 hops
Core Network Hop count	5 hops	1 hop	20 hops
Distance in kilometers	5000 km	1 km	40000 km
MTU of Network	128 bytes	1 byte	2048 bytes

See Figure 5.1 for hierarchical organization of the simulation program algorithm.

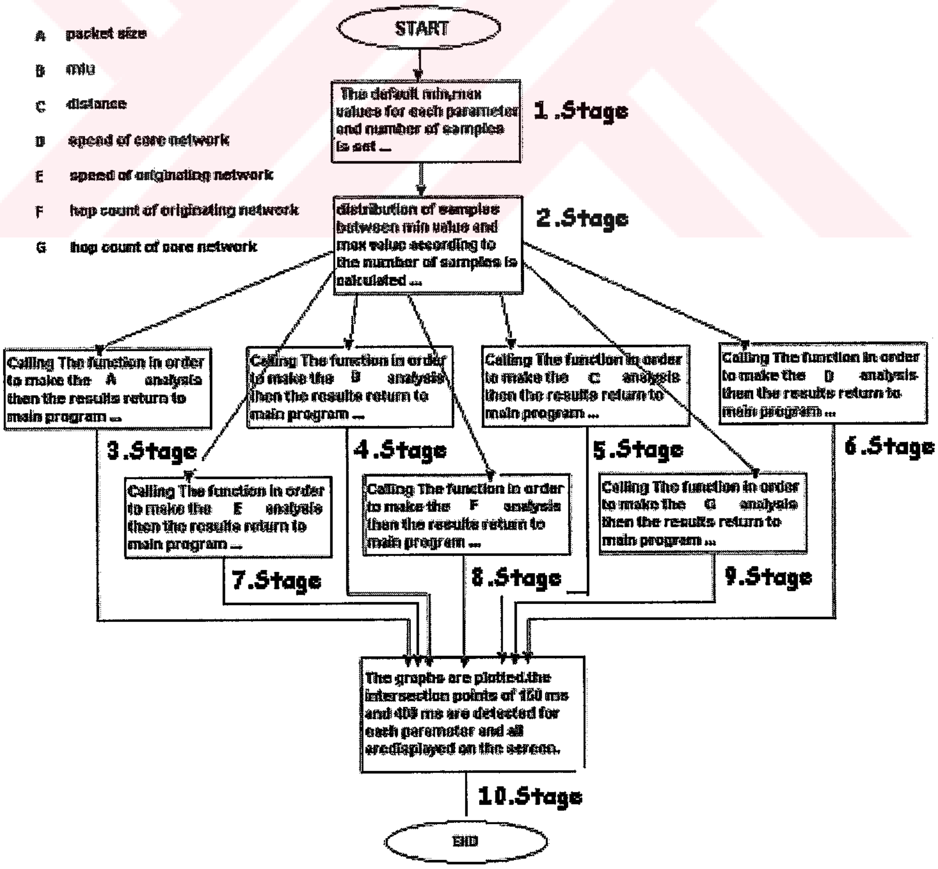


Figure 5.1: Algorithm of the simulation program

5.1 Stages of the algorithm

1.Stage : In this stage the parameter values and the default values (see Table 5.1) assigned for these values are set, the code part of this stage is illustrated below …

```
number_of_samples=100;
psmin_value=1;
psmax_value=300;
psfark=psmax_value-psmin_value;

nhmin_value=1;
nhmax_value=20;
nhfark=nhmax_value-nhmin_value;

scnmin_value=1;
scnmax_value=4096;
scnfark=scnmax_value-
scnmin_value;

nhcmin_value=1;
nhcmax_value=20;
nhcfark=nhcmax_value-
nhcmin_value;
lsmin_value=1;
lsmax_value=4096;
lsfark=lsmax_value-lsmin_value;

dmin_value=1;
dmax_value=40000;
dfark=dmax_value-dmin_value;

mtumin_value=1;
mtumax_value=2048;
mtufark=mtumax_value-
mtumin_value;

packet_length=20;
orij_hops=1;
core_speed=512;
core_hops=5;
link_speed=128;
distance=5000;
mtu=128;
```

where ;

- "number_of _samples" means the number of calculations will be made from minimum value up to maximum value for each delay factor.
- "psmin_value" means minimum value for packet size (bytes)
- "psmax_value" means maximum value for packet size (bytes)
- "psfark" means the difference between minimum value of packet size and the maximum value of the packet size.
- "nhmin_value" means minimum hop count in originating / terminating network
- "scnmin_value" means minimum value for speed (bandwidth) of core network

- “nhcmin_value” means minimum value for hop count in core network
- “lsmin_value” means minimum speed (bandwidth) of originating / terminating link speed
- “dmin_value” means minimum distance between source point and the destination point
- “mtumin _value” means minimum MTU value of the network (bytes)

2.Stage : In This Stage the step sizes (interval of two samples between 2 calculation point) of each factor in the given ranges are calculated and the related functions are called by giving these values to the function. The code part of this stage is illustrated below

```
temp2=linspace(psmin_value,psmax_value,number_of_samples);
```

Here temp2 is a variable that holds the uniformly distributed calculation samples from minimum value up to maximum value in given range.

3,4,5,6,7,8,9. Stages : Each of these stages call their own function and calculate the end-to end total delay amount for that factor. Then the results return to the main function in two arrays, one of these arrays includes the parameter value of that factor used in calculation and the other one includes the total delay amount for the used parameter value. In these functions the calculation is made as illustrated below.

```
Function[min,max]=packet_analysis(number_of_samples,min_value,max_va
lue,orij_hops,core_speed,core_hops,link_speed,distance,mtu,pc)
temp=linspace(min_value,max_value,number_of_samples)
for i=1:1:number_of_samples
packet_length=temp(i);
```

```
look_ahead=5;
encoding_buffer=10;
buffer_delay=10;
pack_delay=packet_length-encoding_buffer;
switching_delay=10*orij_hops;
data_queuing=(((mtu+48)*8)/link_speed);
serializaion_delay=(((packet_length+48)*8)/link_speed);

voice_contention_queuing_delay=2*(serializaion_delay*(((packet_lengt
h+48)/link_speed)/2)*(((packet_length+48)/link_speed)-1));
core_switching_delay=1*core_hops;
core_data_queuing_delay=((mtu+48)*8)/core_speed*(core_hops-1);
core_serialization_delay=(packet_length+48)*8/core_speed*(core_hops-
1);
propogation_delay=distance*5/1000;
term_switching_delay=10*orij_hops;
term_serializaion_delay=(((packet_length+48)*8)/link_speed);
dejitter_delay=10;
decoding_and_decompression=10;
variable_delay=data_queuing+voice_contention_queuing_delay+core_data
_queuing_delay;
total_variable_delay=variable_delay*2;

fixed_total=look_ahead+encoding_buffer+buffer_delay+pack_delay+switc
hing_delay+serializaion_delay+core_switching_delay+core_serializatio
n_delay+propogation_delay+term_switching_delay+term_serializaion_del
ay+dejitter_delay+decoding_and_decompression;

grand_total=total_variable_delay+fixed_total;

if pc==1
Pcdelay=PC_additional_Delays(link_speed,core_speed,packet_length);
        fixed_total=fixed_total+Pcdelay;
        grand_total=total_variable_delay+fixed_total;
        end

min(i)=fixed_total;
max(i)=fixed_total+total_variable_delay;
end
baslik='Packet size in bytes'
draw_result_graphs(min,max,temp,baslik,max_value)
```

In this function the default values of each parameter, minimum and maximum values of the factor that we are making calculations on (e.g. packet size for the given codes) and the pc option (will be 1 if a pc is used and will be 0 if an IP phone is used) are given as inputs of this function, the uniformly distributed step size from minimum packet size value, up to maximum packet size value are calculated values for each calculation are calculated and assigned to the array variable which is named as "temp".

Then the delay amount is calculated for each of these packet size values. If "pc" option is sent as 1 then the additional delays valid for pc implementation are also added to the total delay amount.

10 Stage : Two arrays return to the main function , one holds the packet sizes used in calculation and the other holds the delay amounts calculated using those packet sizes. Then the results are plotted,

All the intersection points of 150 ms and 400 ms with the delay graph are detected and illustrated on the screen by the program at the end of the program. The results of the program are discussed in Chapter 6.

CHAPTER 6

RESULTS and DISCUSSIONS

When the simulation program is run using the default values and observation ranges given in Table 5.1, we obtain the graphs defining the acceptable limits of these parameters. The respective plots are to be found in Figures 6.1 – 6.7. In these graphs, the two bold lines, parallel to the horizontal axes show ETSI defined acceptable and maximum threshold values, which are 150 and 400 ms respectively. Here the region below the lower line would correspond to perfect (best) quality, the region above the upper line would correspond to unacceptable quality, while the area in between would indicate the acceptable region. In these plots, the solid line shows the variations when only fixed delays are taken into account, whereas the broken line curves represent the total delay variations, i.e. fixed plus variable. For the interpretation of our results, we are normally concerned with the latter.

6.1 Packet size analysis

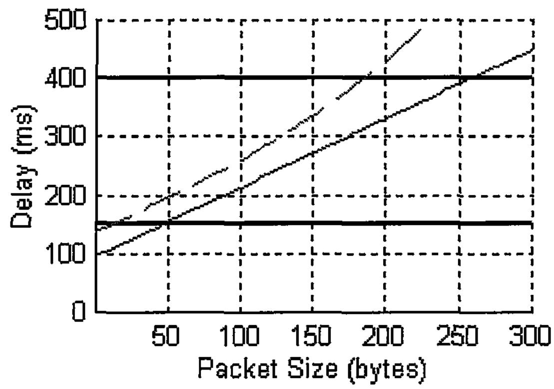


Figure 6.1 : Delay variation versus packet size

In the simulation program, the default packet size is taken as 20 bytes [9] and it is varied from 1 byte to 300 bytes while the other parameters are fixed on their default values. According to the results of this simulation, as also seen from Figure 6.1, the maximum packet size for best quality is 10 bytes and the maximum acceptable packet size is 182 bytes. That means if the average packet size used in our network exceeds 10 bytes, the conversation will not be perfect, but it will have an acceptable quality. But if the average packet size used by the network goes beyond 182 bytes, this conversation will turn into an unacceptable quality of service.

6.2 Originating / terminating network hop count analysis

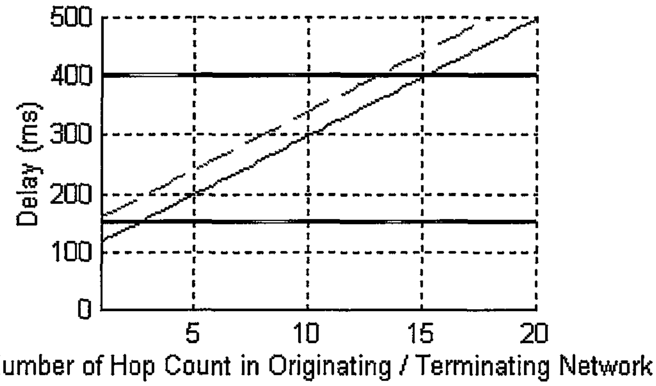


Figure 6.2 : Delay variation versus hop count in originating or terminating network

Number of hops in originating network is defined to be the number of network nodes (routers, switches) that our packet has to pass though before it enters or after it leaves the core network. So it includes the number of hops belonging to originating as well the terminating networks. In the simulation program, the default value for this parameter is taken to be 1 hop and it is varied in the range of 1 hop to 20 hops while keeping the other parameters fixed on their default values. According to the results of this simulation, as also seen from Figure 6.2, the delay value is never in the "best quality range and the maximum acceptable hop count total is 13 hops. That means, with these default values, the conversation will never be perfect, but it will have an acceptable quality level, if and only if the number of hops is not exceeding 13.

6.3 Speed of core network analysis

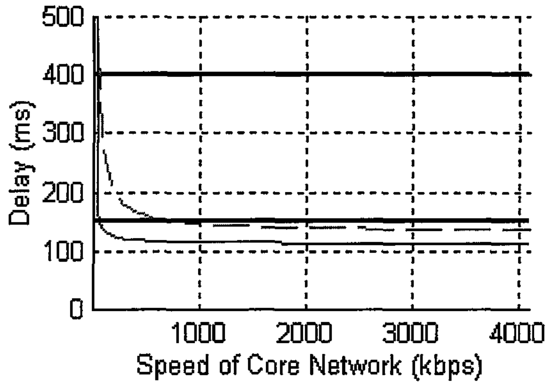


Figure 6.3 : Delay variation versus link speed in core network

In the simulation program, the default value for the guaranteed tunneling bandwidth offered by the core network is taken as 512 kbps and it is varied from 1 kbps to 4096 kbps while the other parameters are fixed on their default values. According to the results of this simulation, as also seen from Figure 6.3, the minimum speed of the core network for best quality is 828 kbps and the minimum acceptable speed is 84 kbps. That means we can have a perfect conversation quality so long as the speed of the core network exceeds 828 kbps and an acceptable conversation quality will be obtained if the speed of the core network does not fall below 84 kbps.

6.4 Core network hop count analysis

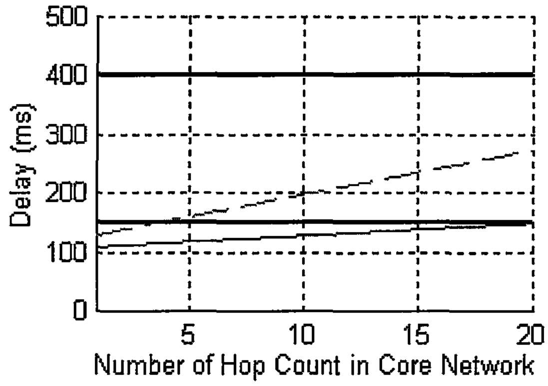


Figure 6.4: Delay variation versus hop count in core network

Number of hops in core network refers to the number of (core) network nodes (routers, switches) that our packet encounters during its travel within this portion of the network. In the simulation program the default value for the hop count in core network is taken as 5 hops and it is varied from 1 hop to 20 hops while the other parameters are fixed on their default values. According to the results of this simulation, as also seen from Figure 6.4, the maximum hop count in the core network for best quality is 4, whereas the maximum acceptable hop count is well beyond 20 hops. That means that almost regardless of the number of hops that the packets pass through in the core network, the conversation will remain within the acceptable quality range.

6.5 Speed of originating / terminating network analysis

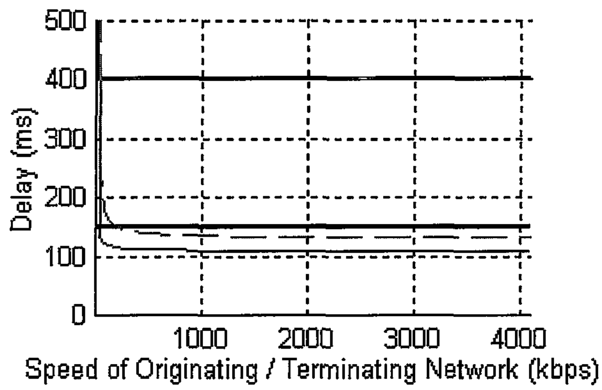


Figure 6.5 : Delay variation versus link speed in originating / terminating network

In the simulation program, the default value for the speed of originating network (and correspondingly of terminating network) is taken as 128 kbps and it is varied from 1 kbps to 4096 kbps while the other parameters are fixed on their default values. According to the results of this simulation, as also seen from the Figure 6.5, the minimum speed of the originating network for best quality is 208 kbps and the minimum acceptable speed is above 42 kbps. That means we can have a perfect conversation quality if the speed of the originating network exceeds 208 kbps and on the other hand an acceptable conversation quality will be obtained provided that the originating network speed is not less than 42 kbps.

6.6 MTU analysis

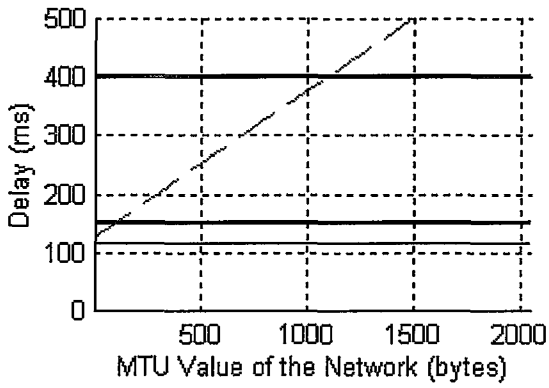


Figure 6.6 : Delay variation versus MTU value of the network

In the simulation program, the default MTU value is taken as 128 bytes and it is varied from 1 byte to 2048 bytes while the other parameters are fixed on their default values. According to the results of this simulation, as also seen from Figure 6.6, the MTU value for best quality is 84 bytes and the maximum acceptable packet size is 1076 bytes. That means if the MTU value used in our network exceeds 84 bytes, the conversation will not be perfect, but it will have an acceptable quality. But if the MTU value that the network uses goes beyond 1076 bytes, this conversation will turn into an unacceptable quality of service.

6.7 Distance analysis

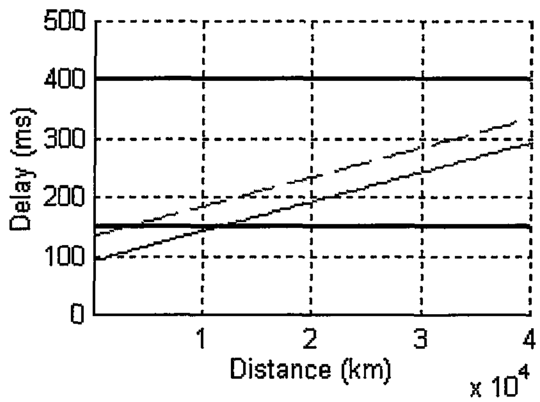


Figure 6.7 : Delay variation versus distance

In the simulation program, the default distance between the source and the destination node is taken as 5000 km and it is varied from 1 km to 40000 km while the other parameters are fixed on their default values. According to the results of this simulation, as seen from Figure 6.7, the maximum distance for best quality is 2829 km and the conversation will always remain acceptable, even when the distance is reaches 40000 km.

Having analyzed all delay parameters independently, we are now in a position to assess the relative contribution of each to the total delay.

6.8 Relative contribution of packet size

As seen from the graph in Figure 6.1, this particular delay increases rapidly with increasing packet size. Therefore it is strongly recommended that we opt for smaller packet sizes. From this graph, it is clear that if we have an average packet size less than 7 bytes, and other input parameters remain fixed on their default values (see

Table 5.1), the conversation quality will never depart from perfection. Once the packet size exceeds 104 bytes, the conversation will become unacceptable. Hence in summary, since any incremental changes in packet sizes cause large increases in delay values, choosing the smallest possible packet sizes would be preferable.

6.9 Relative contribution of hop count in originating / terminating and core networks

As easily observed from Figures 6.2 and 6.4, hop count delay is linearly proportional to the number of hops in originating/terminating and core networks. But in this amount, since the speed of core network is appreciably higher; its contribution is seen to be negligible.

6.10 Relative contribution of speed of originating / terminating and core networks (line speed)

Fundamentally delay is caused by two main items ; distance and speed. Intuitively we may think that minimization of delay may be achieved by reducing the former while increasing the latter. This apprehension is based on the belief that delay is inversely proportional to distance, but directly proportional to speed. Looking at our graphs, we note that this statement holds in general, but eventually a saturation point is reached where increasing the speed any further will no longer serve to decrease the delay. Hence for the effective usage of bandwidth in the network, we have to decide at what optimum speed our originating and core networks should operate.

As all line speeds go to infinity, all speed related delays, i.e. those parameters dependent on originating network line speed, core network line speed and terminating network line speed which are data queuing delay, serialization delay, data voice contention delay, core network serialization delay, terminating network serialization delay and core network data queuing delay, should theoretically reduce to zero. When our simulation program is run with all line speeds set to infinity and all other factors set to their default values, we find that sum of line speed related delays comes to 37.26 ms, where 24.51 ms of this amount is attributed to variable part, while the remaining 12.75 ms is fixed delay originated. On the other hand, the individual delays, that reduce to zero, by setting all line speeds to infinity, are listed in Table 6.1.

Table 6.1 : List of line speed related delays

NAME of LINE SPEED DELAY	Amount (ms)	Delay Type
Data Queuing Delay	11.00	Variable
Data Voice Contention Delay	2.51	Variable
Core Network Data Queuing Delay	11.00	Variable
Originating Network Serialization Delay	4.25	Fixed
Terminating Network Serialization Delay	4.25	Fixed
Core Network Serialization Delay	4.25	Fixed
Total Variable Delay Amount	24.51	Variable
Total Fixed Amount	12.75	Fixed

From these calculations, we note that the minimum total delay that may be attained under the most favorable conditions, i.e. when all speed are at infinity, is 105 ms. This is obtained by subtracting 61.77 (jitter buffer doubles the variable delay) ms from the 166.76 ms which are line speed related and total delay amounts respectively.

6.11 Relative contribution of MTU (Maximum transferable unit)

MTU is the maximum packet size that the network can handle at a time. So we may assume at first sight that the more the number of bytes that a network can transport in one single move, the less will be the delay. But we see the contrary in Figure 3.3, such that the delay increases while the MTU increases. This may be explained by the fact that as the MTU Figure is increased, the network starts facing more data and voice contentions. Hence increasing the MTU value will also lead to increases in total delay. Figure 6.6 also demonstrates that fixed delay is not MTU dependent at all.

6.12 Relative contribution of distance

The simple physical formula, distance = speed x time or time = distance / speed tells us the time that passes during propagation is directly related to distance between source and destination. This quantity is inversely proportional to speed of propagation. But since the propagation speed is too large (200000 km/sec), the distance dependent delay will have a slight effect on the total as seen in Figure 6.7.

CHAPTER 7

COMPARISION OF DELAY AMOUNT OF IP PHONE & PC FOR VoIP APPLICATIONS

Today most people are already using a PC in their home or in their office. This means, that the corresponding delay calculations should be made for PCs as well. In order to compare the delay amounts of VoIP for PC and IP phone, we have to produce delay graphs for VoIP on PC applications too. To this end another simulation program is written including the extra delay parameters for a PC which are ingress delay, egress delay, operating system delay, sound card voice processing delay and modem delay.

7.1 Ingress delay

The time that passes by feeding the bits into the network at ingress point of the core network is called ingress delay and can be calculated as [2];

$$\text{Ingress delay} = \frac{\text{packet length (bytes)} * 8}{\text{Link speed (kbps)}} \tag{7.1}$$

7.2 Egress delay

The time that passes by getting the bits from the network at egress point of core network is called egress delay and can be calculated as [2];

$$\text{Egress delay} = \frac{\text{packet length (bytes)} * 8}{\text{Core speed (kbps)}} \qquad (7.2)$$

7.3 Sound card voice processing delay :

Sound card voice processing delay is the time passes for the sound card to process the voice samples. This processing time creates some extra delay from 20 ms to 100 ms [2].

7.4 Modem delay

The time that passes while the modem modulates/demodulates the voice packets requires an additional time from 20 ms to 40 ms which is called modem delay.

7.5 Comparison of delay amounts for VoIP using IP phone or PC

By adding these extra calculations for each of the delay factors, we have obtained graphs which are different from the previous ones. The graphs are drawn in a super imposed manner, it is easier to see how the delay varies if PC is used instead of IP Phone. Next we investigate how the total delay amount change according to.

7.5.1 Comparison of IP phone and PC for packet size

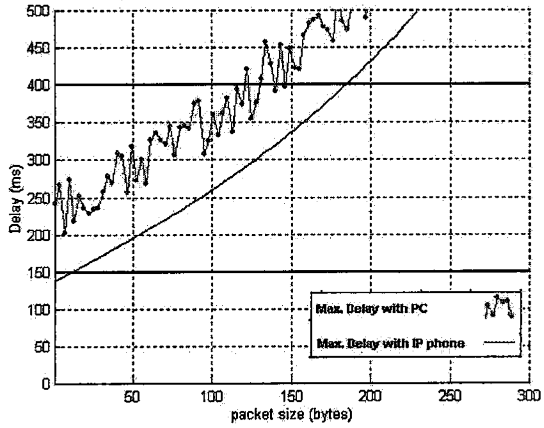


Figure 7.1: Comparison of IP phone and PC for packet size

From Figure 7.1, we see that maximum delay of 400 ms is exceeded when packet size is above 137 bytes for PC phone .The same threshold for an IP phone is 182 bytes. This creates a significant difference of 25 % telling us that we should prefer using IP phones.

7.5.2 Comparison of IP phone and PC for number of hops in originating network

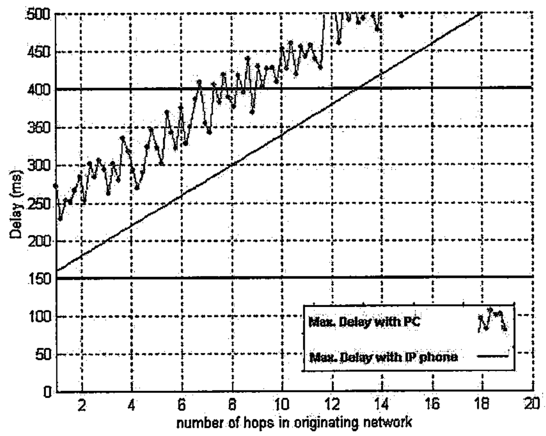


Figure 7.2: Comparison of IP phone and PC for number of hops in originating Network Comparison of IP phone and PC for packet size

From Figure 7.2, we see that maximum delay of 400 ms is exceeded when hop count of originating network is above 9 hops for PC phone .The same threshold for an IP phone is 13 hops. This creates a significant difference of 30 % telling us that we should prefer using IP phones.

7.5.3 Comparison of IP phone and PC for number of hops in core network

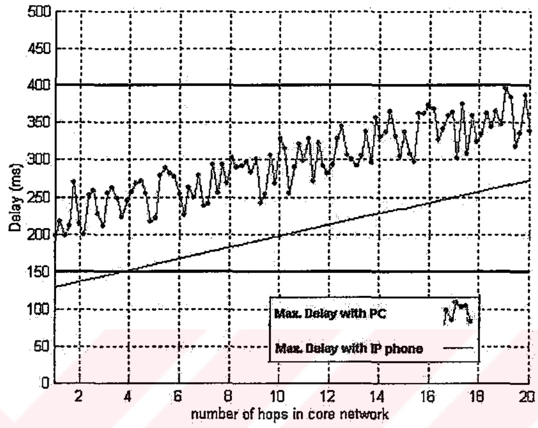


Figure 7.3: Comparison of IP phone and PC for number of hops in core network comparison of IP phone and PC for packet size

As it is easily seen from Figure 7.3 even if 20 hops are used the total delay amount never exceeds the maximum delay figure of 400 ms. But it is to be observer that using IP phone is again preferable.

7.5.4 Comparison of IP phone and PC for core network speed

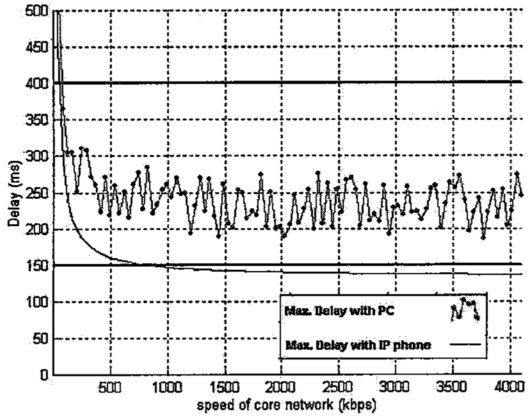


Figure 7.4: Comparison of IP phone and PC for core network speed

By evaluating the outputs of our simulation program shown in Figure 7.4 we note that the minimum line speed in core network should be 84 kbps for IP phone. Whereas the same Figure for a PC phone is 88 kbps.But since this difference is not negligible in high speeds, we should prefer using IP phones instead of PCs for less delay and better quality.

7.5.5 Comparison of IP phone and PC for originating /terminating network line speed

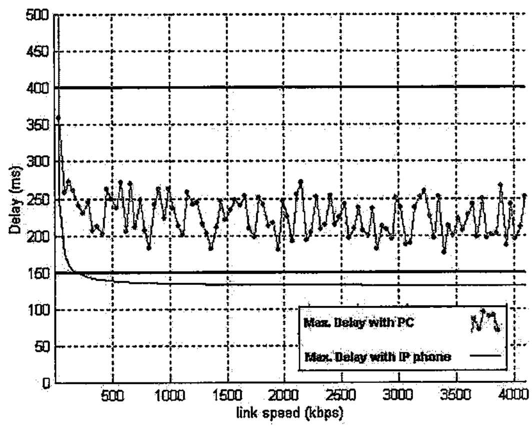


Figure 7.5: Comparison of IP phone and PC for originating /terminating network line speed

From Figure 7.5, we see that maximum delay of 400 ms is exceeded when the line speed of originating / terminating network is below 44 kbps for PC phone. But for an IP phone the delay amount exceeds 400 ms when the line speed of originating / terminating network is below 42 kbps. This again creates a negligible difference in low speeds , but since it has significant difference in high speeds (see Figure 7.5) we should prefer using IP phones.

7.5.6 Comparison of IP phone and PC for distance

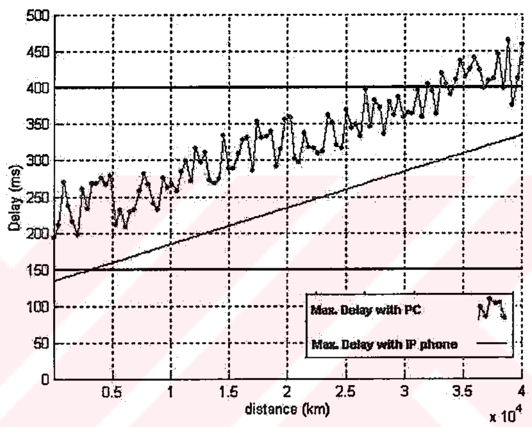


Figure 7.6: Comparison of IP phone and PC for distance

From Figure 7.6, we see that maximum delay of 400 ms is exceeded when distance between source point and destination point is above 39192 kms for PC phone .But for an IP phone the delay amount never exceeds 400 ms even if the distance is 40000 kms. This again creates a significant difference telling us that we should prefer using IP phones.

7.5.7 Comparison of IP phone and PC for MTU

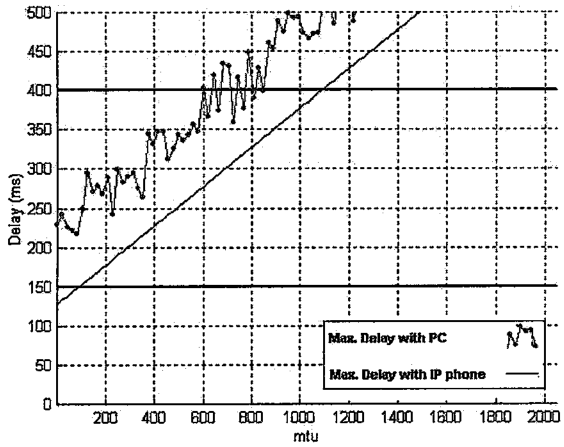


Figure 7.7: Comparison of IP phone and PC for MTU

From Figure 7.7, we see that maximum delay of 400 ms is exceeded when the MTU of the network is above 849 bytes for PC phone .But for an IP phone the delay amount exceeds 400 ms if and only if the network MTU is greater than 1076 bytes. This again creates a significant difference telling us that we should prefer using IP phones.

CHAPTER 8

CONCLUSIONS & FUTURE WORK

In this thesis, we have investigated the various delay factors and the total delay amount that is expected for an IP phone conversation. In relating these factors to the total delay, we have examined each parameter separately and noted that line speed, MTU and packet size play the most decisive roles. We have also observed that having a high line speed is not always favorable option. Increasing the line speed doesn't always decrease the delay amount, because there are some more factors other than line speed that effect the total delay. If we chose large values for the packet size than the delay for real time applications increases, if the packet size is chosen so small, the delay will decrease and quality will increase but this will decrease the reliability of other data packets traversing the network ,so we have to be careful in choosing the optimum value for packet size in our organization. We also should take care about the MTU value of our network and we should choose this value in the determined range in order to have the optimum MTU value for both voice and data packets. On the other hand link speeds for each network should also be chosen at an optimum value which decreases the delay amount down to minimum amount. Because since voice and ordinary data share the same medium, adjusting the MTU, packet size or line speed values only for voice is impossible, because this decreases the reliability and reliability is generally much more important than delay for non real time data packets.

As a summary we can say that, since voice and other data packets share the same medium, providing a high level of QoS for VoIP is really very difficult, because the conditions that increase the quality of conversation decreases the reliability of other data packets.

On the other hand ,results of our simulation indicates and we have been proven that, since PC implementation of VoIP has some additional delays such as modem delay, sound card processing delay or etc., using IP phone is always favorable than using PC for less delay and better quality.

Future Work: In practical VoIP implementations, since most people would tend to use PC instead of an IP phone at their homes or offices, the subject of our future work will be determining if the performance of IP phone on VoIP implementations, can also be provided by PC's or not.

REFERENCES

[1] Eyyuboğlu, H.T. "A Wide Angle Perspective of Telecoms World" , Eleco2003 Conference, 3-7 December 2003, Bursa, Turkey.

[2] Wright, D.J. *Voice Over Packet Networks*, John Wiley & Sons, Ltd. ,2001.

[3] Cisco Systems White Thesis. "Understanding Delay in Packet Voice Networks" (2003) from http:// www.cisco.com/warp/public/788/voip/delay-details.pdf.

[4] ITU, "G.114 Delay Recommendations" , 1996.

[5] ETSI TR 101 329-7 2.1.1 "Telecommunications and Internet Protocol Harmonization over networks release 3: End to end quality of service in TIPHON systems Technical Report" 2002-02

from webapp.etsi.org/action\PU/ 20020226/tr_10132907v020101p.pdf

[6] Brans, T., De Keyser, T., Peirs, C. and Pollin, S. , "Voice Over IP " , December 2001. from www.esat.kuleuven.ac.be/~h239/ reports/2001/voip/verslagvoip.pdf

[7] Cisco Systems White Thesis, "Understanding Codecs: Complexity, Hardware Support, MOS, and Negogiation", 2002

from www.cisco.com/warp/public/ 788/voip/codec_complexity.html

[8] Metha, P.C. and Udani, S. "Overview of Voice over IP Technical Report MS-CIS-01-31" , 2001 . from http://citeseer.ist.psu.edu/mehta01overview.html.

[9] Preveze,B. and Eyyuboğlu, H.T. "A simulated Analysis of Delay factors for VoIP" , Accepted paper for Conference IKS-2004, Asos - Turkey, August 2004.

[10] Black, U. *Voice Over IP* , Prentice Hall PTR , 2001

[11] Held, G. *Understanding Data Communications*, John Wiley & SONS Ltd, 2000

[12] Beritelli, F. ,Casetti, C. ,Giordano, S. , R.L. Cigno , "Overview of the research results on Voice over IP"

from www1.tlc.polito.it/mqos/courmayeur/sorgenti/sor3.pdf

[13] Baldi, M. ,Rissio ,F. , "Comparing the Efficiency of IP and ATM Telephony" from netgroup.polito.it/fulvio.risso/pubs/icatm99-pres.pdf

[14] Goyal, P. ,Greenberg , A., Kalmanek, Charles R. ,Marshall, W.T., Mishra, P., Nortz D., Ramakrishnan, K.K , "Integration of Call signaling and Resource Management for IP Telephony". from www.komunikasi.org/acrobat/vop/Integration_of_Call_Signalling_and_Resource_Management_for_IP_Telephony.pdf

[15] White Thesis , "Some aspects concerning IP-Telephony"

from www.digit-life.com/articles/aspectsiptelephony