

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY**

CAMPUS MONTERREY

**DIVISION DE TECNOLOGIAS DE INFORMACION
Y ELECTRONICA**

PROGRAMA DE GRADUADOS EN INGENIERIA



**SIGNAL DEGRADATION DUE TO DELAY STATISTICS
IN VOIP NETWORKS**

THESIS

**Presented as a partial fulfillment of the requirements
for the degree of**

**Master of Science in Electronic Engineering
Major in Telecommunications**

Fernando Sánchez Durán

Monterrey, N. L. May 2007

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Campus Monterrey

Escuela de Tecnologías de Información y Electrónica

PROGRAMA DE GRADUADOS



SIGNAL DEGRADATION DUE TO DELAY STATISTICS IN VOIP NETWORKS

THESIS

Presented as a partial fulfillment of the requirements for the degree of

Master of Science Electronic Engineering

Major in Telecommunications

Fernando Sánchez Durán

Monterrey, N.L. May 2007

SIGNAL DEGRADATION DUE TO DELAY STATISTICS IN VOIP NETWORKS

Fernando Sánchez Durán

THESIS

Presented as a partial fulfillment of the requirements for the degree of
Master of Science Electronic Engineering
Major in Telecommunications

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY**

May, 2007

Abstract

The present work develops a method to measure the Quality of Service (QoS) perceived by the user, two approaches are given. First is presented the Telephone Number Mapping (ENUM) protocol, with some details for the implementation of the provisioning system and a mobile client example, as an external aid to improve the perceived quality of the service perceived by the user. This document concludes with a packet layer analysis to describe the Degradation induced by the state of the network in the perceived QoS in a VoIP communication. The results were simulated for normal and exponential R.V and a reparametrization in the state of the network parameters (jitter) is done by an end buffer. A viable method to model real time scenarios is included with the facilities to compute a estimation of the Mean opinion Score (MOS) through recommendation ITU-T P.862.

Contents

CHAPTER I INTRODUCTION	9
1.1 Objective	10
1.2 Justification	12
1.3 Summary	13
CHAPTER II ENUM as a VoIP QoS booster [29]	14
2.1. What is ENUM?	14
2. 1. 1. ENUM protocol Functional description	15
2.2. Client – Server Architecture	16
2.3. Provisioning System ITESM-NIC ENUM Trial Mexico	16
2.4. ENUM client in mobile devices	20
2.5. ENUM queries impact on DNS servers	21
CHAPTER III SPEECH MEASURMENT BACKGROUND	24
3.1 Antecedents measuring voice quality	24
3.2 . Quality assessment strategies	25
3. 2. 1. Subjective quality assessment	25
3. 2. 2. Objective quality assessment	26
3.3 Types of measurements techniques	26
3. 3. 1 Intrusive (active)	26
3. 3. 2 Masked-error models	27
3. 3. 3 Models based on comparison of auditory transforms	28
3. 3. 4 Models for network testing	29
3.4 Non-intrusive (passive)	31
3. 4. 1 Passive method 1	31

3. 4. 2	Passive method 2	34
3.5	Methods for network testing (Analysis of network performance)	36
CHAPTER IV NETWORK MEASUREMENTS AND IMPLEMENTATION OF VOIP MONITOR		40
4.1.	Issues to consider when designing a VoIP QoS monitor	40
4.2.	The configuration parameters of periodic sampling	41
4. 2. 1	Parameters collected at the measurement point MP (Source LAN)	42
4. 2. 2	Parameters collected at the measurement point MP (Destination LAN)	42
4.3.	Metrics to sample	42
4. 3. 1	One-way-Delay Metric [14]	42
4. 3. 2	IP Packet Delay Variation Metric or Jitter (ipdv) [15]	44
4. 3. 3	One-way Packet Loss Metric [16]	45
4.4.	RTP: A Transport Protocol for Real-Time Applications [19]	45
4.5.	Implementing a VoIP QoS Monitor	47
CHAPTER V RESULTS		50
5.1	Ideal scenario for packetized discrete tones stream	52
5.2	Poisson Process	55
5.3	De-jittering and Lost Packets Inclusion	57
5.4	Introducing Correlation and Power Spectral Density for (5.1)	59
5.5	Power Penalties Results	64
5.5.1	Gaussian distributed R.V power penalties	64
5.6	Power Penalties analysis with a de-jitter buffer inclusion	68
5.7	Linewidth analysis	71
5.8	Linewidth analysis with a de-jitter buffer inclusion.	73

Figures

Figure 1.1: Testbed and VoIP QoS Monitor.....	11
Figure 2.1: Provisioning system ENUM Protocol.....	1
Figure 2.2: CDMA-SIP call using ENUM protocol.....	19
Figure 2.3: Mobile ENUM client example for 3G Devices.....	21
Figure 3.1: VoIP degrading quality of service impairments.....	25
Figure 3.2: Illustration of an intrusive test.....	27
Figure 3.3: ITU – T P. 862, Perceptual Evaluation of Speech Quality (PESQ) block diagram.....	30
Figure 3.4: Passive method for objectively measuring the subjective voice quality of live VoIP calls.....	32
Figure 3.5: Sliding of a speech sample along a packet trace. “A” and “B” denote important and unimportant packets, respectively.....	34
Figure 3.6: Testbed.....	37
Figure 4.1: RTP Fixed Header Fields.....	47
Figure 4.2: VoIP QoS Monitor.....	48
Figure 5.1: Total Latency contributors.....	51
Figure 5.2: 8.2 seconds frequency tone, (a) Original signal and, (b) Packetized signal.....	53
Figure 5.3: System $h(t)$ processing the input $Z(t)$ to generate the output of the form (5.1).....	56
Figure 5.4 A better approach for a real VoIP system communications. See Equation 5.9.....	57
Figure 5.5: Simulation of a De-jitter buffer for an UDP/IP/RTP stream affected by modulation random phase noise.....	58
Figure 5.6: Autocorrelation function for an exponential R.V. with parameter.....	60
Figure 5.7: Autocorrelation function for an exponential R.V. with parameter.....	61

Figure 5.8: Autocorrelation function for an exponential R.V. with parameter..... 61

Figure 5.9: Autocorrelation parameter h dependence on the exponential R.V delay mean..... 62

Figure 5.10: Autocorrelation parameter h dependence on the exponential R.V mean delay, this zoom shows the independence of the autocorrelation and the R.V. for very small means 63

Figure 5.11: Power spectral density comparison, exponential distributed delay with parameter and 5 tones duration. The tones lengths are specified at legend in figure. Here there is tones centered at 1477 Hz 63

Figure 5.12: Power penalties for a DMTF signal with an exponential R.V mean delay in the range (5.11). No de-jitter buffer was considered in the simulation. . 65

Figure 5.13: DMTF signal at frequencies 697 and 1477 Hz. Exponential mean 13 tones lengths specified at the legend in figure. The largest tone duration in time, the lowest power penalties it has..... 65

Figure 5.14: Power penalties comparison for a tone of 500ms length, mean delay normal distributed, and 14 standard deviations (1e-7 5e-7 1e-6 5e-6 1e-5 5e-5 1e-4 5e-4 1e-3 5e-3 1e-2 5e-2 1e-1 5e-1). The highest losses are for standard deviations larger than 1 ms. 66

Figure 5.15: Power penalties comparison for a tone of 500ms length, mean delay normal distributed, and 14 standard deviations. An oscillation of less than 2 dB in Power penalties is appreciable for mean delay range of , but less determinant compared with the losses incurred by delay variation. 67

Figure 5.16: Power Penalties transition with the increment of parameter σ in the range (5.14)..... 67

Figure 5.17: Power Penalties comparison for an exponential distributed R.V in the range (5.11). The dotted line with specifications at legend has the best Power Penalties..... 69

Figure 5.18: Power Penalties comparison for an exponential distributed R.V distributed in the range (5.15). The SQT factor does not have any considerable effect on Power Penalties..... 69

Figure 5.19: Power Penalties from a stream of packetized fragments exponentially delayed transmitted tone with mean delay R.V ranging in (5.14). 70

Figure 5.20: Bandwidth against exponential delayed plot for a tone of 8.2 s length implemented without buffer. The plot shows how the bandwidth of the pulse grows for exponential means below 10ms, then the pulse bandwidth remains constant in 45 Hz for exponential means above this value..... 71

Figure 5.21: Line-width tone variation for different tone lengths Fragmented and Packetized. Exponential R.V with parameter $1/\lambda$ in (5.11). It can be seen the rapid dispersion of the pulses along the spectrum. A system like this would be useful only for mean delay parameter $1/\lambda < 1^{-4}$ seconds..... 72

Figure 5.23: The cleanest and noisiest power spectral density signals plotted above keeps the original transmitted pulse bandwidth. The pulses correspond to tone lengths in time of 32 ms and 2.048 s, respectively. 73

Figure 5.25: Mean packet Delay playtime after buffer 76

Figure 5.26: A better approximation of the Mean packet playtime to observe the 100-150 ms zone..... 76

Figure 5.27: Mean packet playtime jitter for an originally exponential R.V with parameter (5.15)..... 77

Figure 5.28: A better approximation of the Mean packet playtime jitter to observe the 100-150 ms zone..... 77

Figure 5.29: Mean Packet Losses for an exponential distributed R.V with mean delay in the range (5.15)..... 79

Figure 5.30: Mean Packet Losses for an exponential distributed R.V with mean delay..... 79

Figure 5.31 Mean Delay variation for an exponential distributed R.V with mean delay in the range (5.15)..... 80

Figure 5.32 Mean Delay variation for an exponential distributed R.V with mean delay in the range (5.15)..... 80

Figure 5.33: Number of Packet Losses dependence with Buffer size..... 81

Figure 5.35: Mean Delay Jitter Playtime dependence with SQT. 83

Tables

Table 2.1: Impact of 5 queries/second on the CPU for ITESM campus Monterrey recursive DNS server with DNSEC implemented.	22
Table 3.1: Correlation coefficient, 8 unknown subjective tests (PESQ).....	31
Table 3.2: Error distribution, 7 unknown subjective tests (PESQ). Test 3 excluded as data was per-file only.....	31
Table 3.3: Means and standard deviations for the ULP for different number of web-users.....	36

CHAPTER I INTRODUCTION

Telecommunications worldwide has experienced a significant revolution over recent years. The long-held promise of network convergence is occurring at an increasing pace. This convergence of data, voice, and video using IP-based networks is delivering advanced services at lower cost across the spectrum, including residential users, business customers of varying sizes, and service providers.

One of the key technologies driving this convergence is VoIP (voice over IP), which has evolved from what many viewed as experimental to a fundamental technology on which businesses from small to multinational are running their enterprises. VoIP has moved to a level of reliability and capability such that mainstream users are adopting it at a rapidly increasing pace. For this to happen, a number of technical innovations were required to solve issues such as quality of service and reliability.

The fundamental concern for VoIP QoS (quality of service) is speech quality. Unfortunately, objective measurements for this have been elusive. That said, the major factors that affect voice quality are delay, packet loss, jitter, and treatment at the endpoints.

Voice codecs are unevenly tolerant of packet loss, but in general, a loss above 2 to 5 percent will have a perceptible effect on quality. Loss is rarely random and is often associated with high jitter defined as an estimate of the packet arrival time variation at the destination).

When one-way delay through a voice network exceeds about 150 milliseconds, natural conversational communication is strained, so most network deployments attempt to keep the delay well below that threshold.

In recent years, we've seen increasing adoption of VoIP networks for customers of varying sizes on a global basis. The cost advantage resulting from convergence and the value of new applications offered by this convergence are the primary drivers of this adoption. With this comes the need for increasingly sophisticated systems and management tools to allow for the extensive adoption and deployment of VoIP.

VoIP's increasing adoption will have a significant impact on our communications and the products that provide those communications. Therefore, software developers across the industry will increasingly need to be aware of and understand the challenges that come with this latest change in the communications infrastructure.

A number of exciting new services and concepts are coming out of the VoIP community. We highlight just a couple as follows: the impact of IM (instant messaging) and presence on converged communications; and ENUM, a mechanism for telephone number resolution in VoIP networks. ENUM has significant impact on Networks because it helps to improve the perceived QoS by the user of any Network, being its adoption necessary for any region in world as discussed in next chapter.

1.1 Objective

Develop a method able to describe the physical degradation of audio signals in networks and applications of VoIP. The resulting method must relate the impairments causing the physical degradation of the received audio signal to the state of the network. The state of the network shall be represented by a random

inter-arrival time of packets forming the audio media stream and the packet losses as a result of collisions and delays overflowing a defined quality threshold.

The difficulty of describing accurately the distribution of the network traffic and its relation with the delay inter-arrival variability distribution of the packets forming the audio media stream is not the only problem to be solved. The frequently evolution in size and kind of byte streams carried by IP networks is a clear insight of the necessity and importance of sampling in real time the given network, see Figure 1.1, this updated statistical information shall be used to describe the physical distortion of the audio signal at the receiver and provide measures of the impairments affecting the signal.

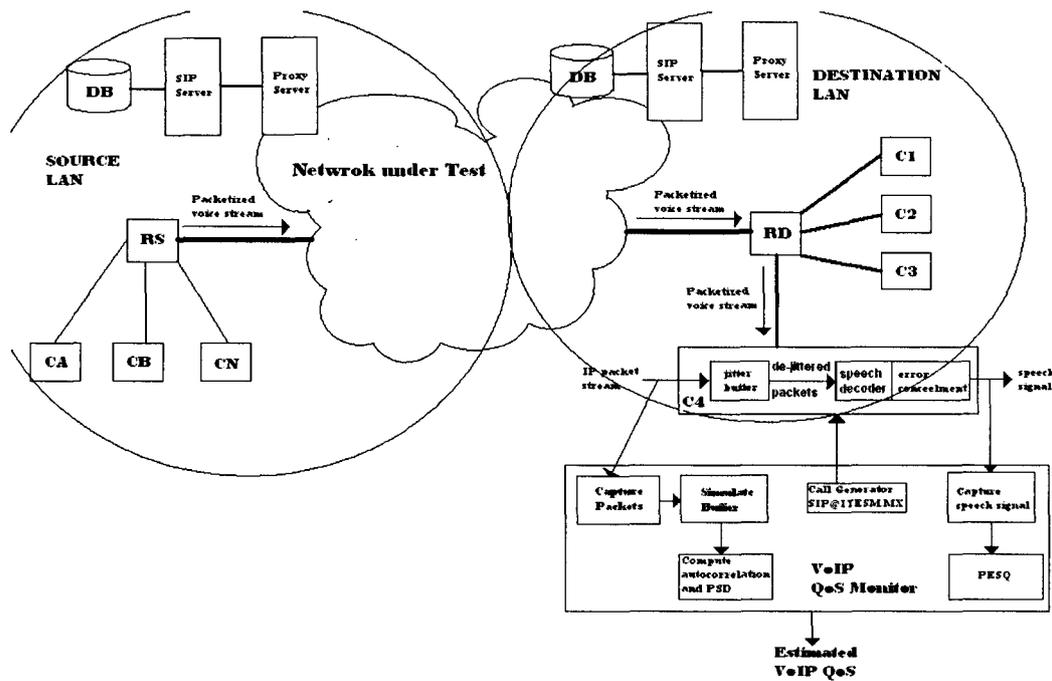


Figure 1.1: Testbed and VoIP QoS Monitor

Figure 1.1 describes graphically a process for sampling in real time a given network; the traffic can also be simulated using a traffic generator to measure the system under stress. The VoIP QoS analyzer of the Figure 1.1 is built by 4 main functions: Call generator, Packet Sniffer, Data statistics processor and Dejitter Buffer. Each block will be discussed later in chapter IV and V, for the specification on how to program a VoIP monitor and a de-jitter buffer, respectively.

1.2 Justification

VoIP is a technology being adapted rapidly at the enterprise, home and all sectors of modern life, mainly because the low cost it represents compared with its homologous Networks (e.g., PSTN and Cellular Network). Nevertheless as an emerging technology, issues regarding the control and monitoring of quality are actual necessities to be satisfied, in order to impulse the VoIP technology.

In [29], the evolution of technologies is described focusing in the transportation of speech over large distances; the technologies are in three major groups: Analog transmission, digital transmission, and packetized transmission of a voice stream. Due to the recently massive adoption of this technology, it is bringing out the problematic of modeling accurately the impairments degrading the quality of the VoIP service on a given network. Basically, the problem is the diversity and complexity of measurements and statistics available describing penalties incurred over IP telephony due to the nature of its transmission.

The growing demand of VoIP services, the large investment in technology deployment, and the growing innovative services around this technology (e.g., ENUM protocol) are enough motivation to invest resources to fully characterize the effect of impairments in VoIP networks in order to establish requirements that would bring speech and service quality to similar levels to those observed in traditional voice networks.

1.3 Summary

The Thesis is organized in the following manner:

Chapter II: This chapter gives details to improve the perceived quality of service by the user of any given network through a provisioning telecommunications identifiers system operating under ENUM protocol and clients making use of this technology

Chapter III: This chapter presents the basic concepts for VoIP analysis of objective and subjective measurements for QoS.

Chapter IV: In this chapter, the concepts needed and the tool requirements to start monitoring a network offering VoIP services are detailed. With this information, it is possible to model the state of the sampled Network and evaluate its capabilities to provide IP telephony services.

Chapter V: This chapter presents a method to model the penalties incurred over a VoIP transmission. All the results obtained make use of the statistics of the given network obtained directly from the information carried by each packet. Graphs were generated from where it is possible to understand the state of the network impact on the quality of the packetized voice stream.

CHAPTER II ENUM as a VoIP QoS booster [29]

The ENUM protocol is known as the directory of the next generation network, NGN. Some implementations around the world such as those of Austria and Germany can be taken as an example to analyze and infer from the statistical data available at the moment, the impact of ENUM queries on the DNS infrastructure.

This chapter explains the implementation and maintenance of the ENUM protocol as well as some clients that already are proposing services and applications based on this protocol. The study is further complemented by an impact analysis of ENUM queries on DNS servers.

Under realistic scenarios, where VoIP, PSTN, GSM, etc. calls massively coexist, the user overall perception of quality is expectedly deteriorated if there is no way to dial a VoIP line knowing only the telephone number associated to the called party. ENUM solves this problem and the inclusion of this chapter is intended to describe the implementation of the ENUM protocol in the Mexican NIC-ITESM trial.

2.1. What is ENUM?

ENUM is a protocol standardized by the Internet Engineering Task Force (IETF), the organization responsible for creating the standards that ensure a smooth operation of the Internet. ENUM is defined in RFC 3761. Basically, this protocol proposes E.164 telephone numbers as the universal and unique identifiers for all telecommunication and network addresses. In essence, once a telephone number is associated to an identifier or to several identifiers in a global

database, they can be extracted by querying the number to the database. ENUM provides then the means to converge all service addresses and identifiers on telephone numbers. This directory can be accessed from any point of the world with an Internet access.

2. 1. 1. ENUM protocol Functional description

ENUM has a client/server architecture that uses the DNS given that it is a globally distributed database. The protocol defines the steps necessary to implement an ENUM query in an ENUM client:

1. Users Dials Telephone number within an ENUM-enabled application or device:

+528183875346

2. The ENUM client removes the “+” sign and adds dots between digits:

5.2.8.1.8.3.8.7.5.3.4.6

3. The ENUM client inverts the number:

6.4.3.5.7.8.3.8.1.8.2.5

4. The e164.arpa domain is added. The resulting domain is called an ENUM domain:

6.4.3.5.7.8.3.8.1.8.2.5. e164.arpa

5. And the DNS query is executed.

Example:

```
>>dig 1.0.0.0.2.5.enum.org.mx naptr
```

```
>>IN NAPTR 10 100 "u" "E2U+sip" "!.*${sip:glozano@sip.enum.org.mx!" .
```

2.2. Client – Server Architecture

The operation of ENUM requires three basic processes: queries, responses, and registration. As stated in the previous section, the ENUM client is in charge of carrying out queries. Typically, an ENUM client takes the form of an API installed in a communication application, such as chat, web browsers, mail, sip phones, etc. The DNS server is in charge of answering petitions since queries include an ENUM domain and the identifiers associated to it are stored in the *Naming Authority Resource Record* field of the DNS. DNS has several qualities that make it appropriate for such an application; to name a few, it is a distributed open access database, it is scalable and reliable, and already installed and in operation worldwide. The DNS architecture consists in an open access distributed database that makes it a vulnerable system to many kinds of security attacks. These attacks are already being studied and solutions against these threats are being implemented [25]. The third process, registration, is usually carried out by companies licensed to authenticate owners of E.164 numbers and modify DNS records called registrars.

2.3. Provisioning System ITESM-NIC ENUM Trial Mexico

On January, 2006 the provisioning system ENUM Trial-Mexico started to operate as a Database formally in production. The range for trial ENUM-numbers assigned in the first stage of trail was:

5200000 – 5299999 (2.1)

The implementation for the ENUM Mexico-Trial provisioning system is depicted in Figure 2.1; One of the most important issues about the implementation is the validation of the user session in every stage the user moves on the provisioning system, this shall help prevent the user identity to be misused by a man in the middle between the provisioning system and the server supporting it.

An ENUM provisioning system should be composed at least by the next components:

- a. Signing in: The user registers a valid username and password to get access the provisioning system services.
- b. Registrar confirmation: The user must confirm its registration by clicking on a registration number sent him by email.
- c. Logging in: The user validates its identity using a Username and Password.
- d. Password recovery: Give the user a temporary password if he proves to be the owner of an ENUM account.
- e. Assigning a (Trial) ENUM number to user: This assigned numbers shall be used by the user to store its telecom identifiers.
- f. Provide a friendly interface to help user manage his ENUM numbers; here the user can perform the next actions:
 - i. Update the stored information in the user's ENUM number.
 - ii. Deletes some or all the information stored by the user's ENUM number.
 - iii. Add a new identifier to be stored in the user's ENUM number.
- g. Provide the user with a list of all his assigned ENUM numbers and the identifiers stored by each.
- h. Provide the user with an ENUM directory (e.g., ENUM lookup) able to make a query under the *trial domain* or *e164.arpa* domain.

Each stage is explained in more detail at Figure 5.1 which provides a block diagram to facilitate the compression of a provisioning system operating under ENUM protocol.

Figure 5.2 shows a VoIP-CDMA call, where a hand-off from 2 GHz frequency for CDMA to a frequency among 2-11GHz band for WiMAX.

Here there are 2 components using or operating under ENUM protocol:

1. SIP server supporting ENUM lookups, like the open source PBX Asterisk or SER (SIP Express Router). This server makes a query to DNS, generating traffic of 800 bytes per query; the DNS will answer with 6000 bytes back.

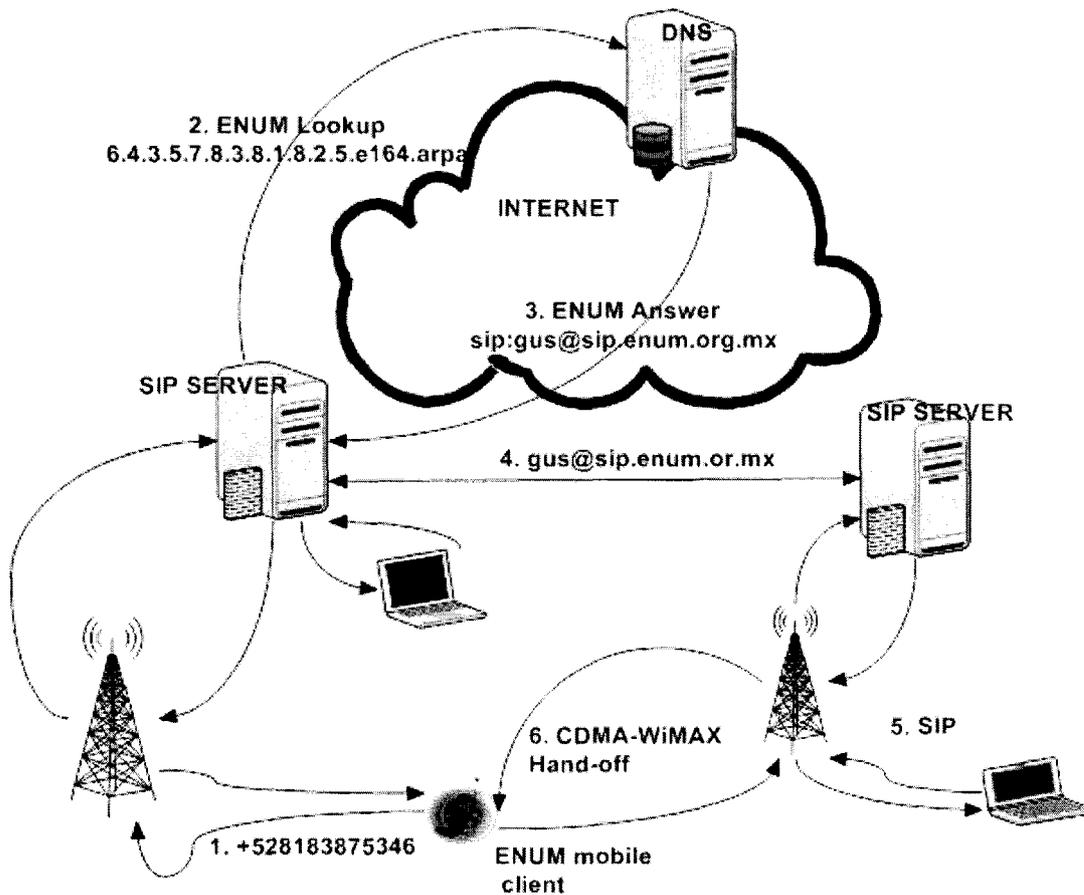


Figure 2.2: CDMA-SIP call using ENUM protocol

Here is where in some cases this situation could present, if the a Source1 does not have in hand the identifiers to initiate VoIP communication with Denstiny1, and neither exists a provisioning system storing this Telecom ID's, it does not matter if Source LAN and Destiny LAN both have delay variation playtimes below $1E^{-4}$ seconds, as it will be discussed the next chapter.

From here the importance for any region to have an ENUM provisioning system, because it provides some amount of value to the perceived quality of the system by the user.

2.4. ENUM client in mobile devices

This is an example of what a client using ENUM protocol could be, and as a computer facilitates the calls to start and finish in Internet, a mobile device like a 3G device will promote the calls to start in cellular network and finish it in Internet; a mobile 3G device could promote as well the starting of calls in cellular network and finish them in Cellular or PSTN network.

A block diagram of how to implement a mobile client using the provisioning system described in the last section is shown in Figure 2.3 [24].

Figure 2.3 shows an ENUM client for mobile 3G devices implemented in the programming language BREW (Binary Runtime Environment for Wireless), an application development platform created by [26] for mobile phones. It is air-interface independent, i.e. it can support GSM/GPRS, UMTS, and CDMA. BREW is a software platform that can download and run small programs for playing games, sending messages, sharing photos, etc.

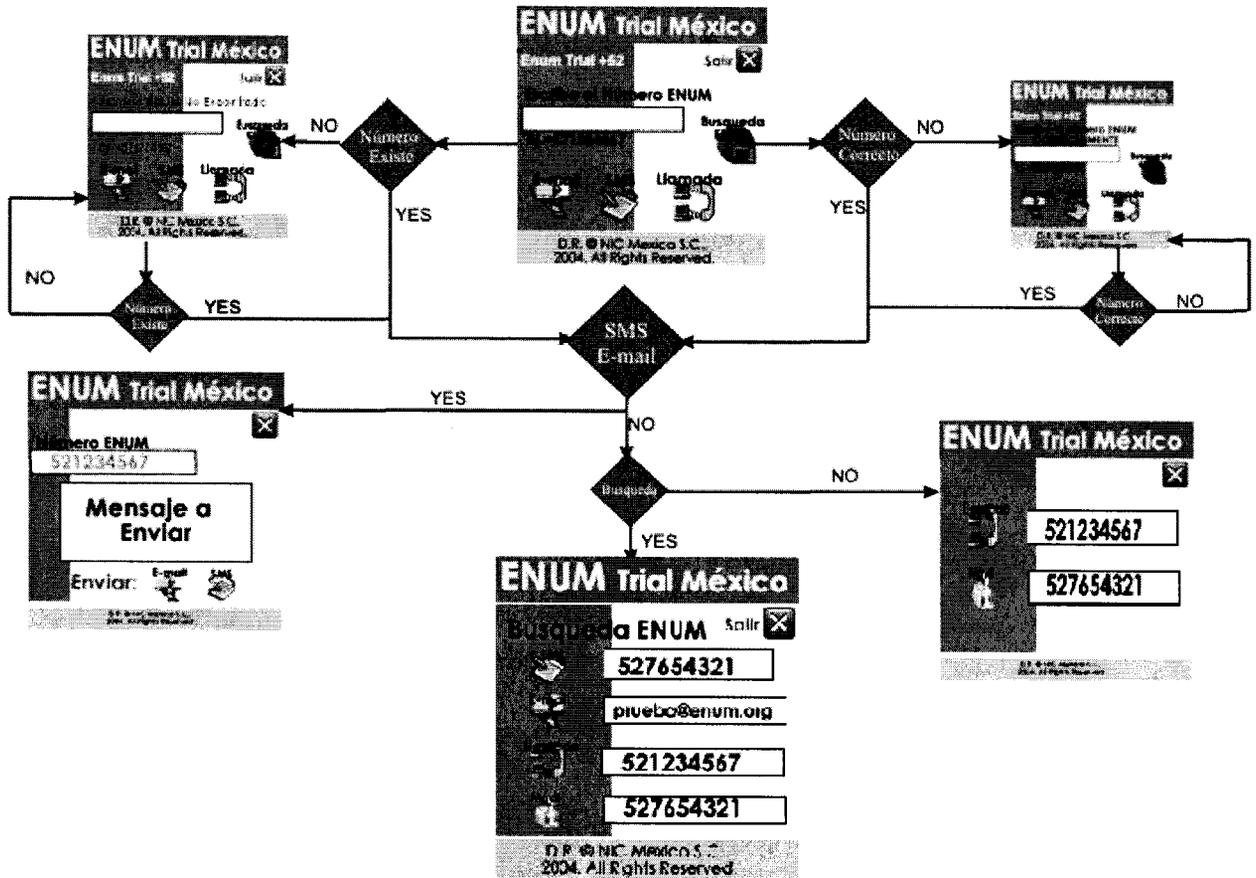


Figure 2.3: Mobile ENUM client example for 3G Devices.

More details of how to create an ENUM client for mobile 3G devices can be found at Appendix B.

2.5. ENUM queries impact on DNS servers

From reference [27] some statistics were obtained dating from June 2006 when the ENUM protocol started to commercially operate in Germany by deNIC, who also manages the domains in London, Amsterdam, Stockholm, Berlin, Frankfurt, Stuttgart, and Vienna.

From the report presented in [22], some graphs showing the distribution of ENUM Lookup traffic to DNS average of 100 queries/minute during peak business hours in three servers, ENUM1, ENUM2, and ENUM3;

This server makes a query to DNS, generating a query of traffic 800 bytes per query, the DNS will answer with 6000 bytes back, in average.

The for a trial that started in 2002 and became commercial by June 2006, considering they started 2 employees in 2002 and by 2004 they had ten times that amount of people working on the project, the ENUM queries traffic generated by 2006 in deNIC servers was:

$$\begin{aligned}
 R_Q &= 3 \cdot 100 \frac{\text{queries}}{\text{min}} \cdot 800 \frac{\text{bytes}}{\text{query}} \cdot 8 \frac{\text{bits}}{\text{byte}} \cdot \frac{1 \text{min}}{60s} = 32\text{kbps} \\
 R_A &= 6000 \cdot 100 \cdot 8 \cdot 3 \cdot 60 / 3600 = 240\text{kbps} \\
 R_T &= 272\text{kbps}
 \end{aligned}
 \tag{2.1}$$

From statistics provided in [27] and [28] we will now compute the impact in the performance for the DNS recursive servers of Tecnológico de Monterrey. The impact for an average of 5 ENUM queries/second at peak hour, in the cpu power processing percentage is resumed in Table 2.1, for a comparison among CPU processing power penalties against length of the key used to sign for DNSSEC protocol.

From Table 2.1 we conclude, the processing power penalties resulting from processing 5 ENUM queries per second, originated by 6100 ENUM domains registered in Germany, are less than 0.833% for key lengths below 4096 bits.

Table 2.1: Impact of 5 queries/second on the CPU for ITESM campus Monterrey recursive DNS server with DNSEC implemented.

Experiment	% CPU
Unsigned	0
512_512	0.333
512_1024	0.333
512_2048	0.333
512_4096	0.333
1024_1024	0.333
1024_2048	0.333
2048_2048	0.333
2048_4096	0.333
4096_4096	0.833

CHAPTER III SPEECH MEASUREMENT BACKGROUND

3.1 Antecedents measuring voice quality

In conventional telephony until 1980, where the speech signal bandwidth was fixed at 0.3-3.4 kHz, the impairment factors were transmission loss, frequency distortion, stationary circuit noise, and, in digital systems, signal correlated quantization noise associated with pulse code modulation (PCM), and so on. These properties are ultimately described by signal-to-noise ratio (SNR). Since VoIP systems are based on new technology (voice over packet-switched networks that were originally designed for non real-time data, e.g. IP+RTP+UDP+SIP), the primary determinants of the perceptual QoS of a VoIP systems are distortion caused by speech coding and packet loss, loudness, delay (network and terminal delay), jitter, and echo [1]. These and other impairments affecting the quality of an audio signal are shown at Figure 3.1.

The convergence of telecommunication networks recently approved in Mexico, also known as triple play, stands for the legal offering of voice, data, and video services by any licensed telecom network operator.

The measuring and design of convergent networks must include statistics about the population and geography where the network will operate, the total input and output traffic and the kind of data being transported. Traffic can be classified as internal and external web traffic, databases querying and update traffic, internal and external email traffic, terminal traffic and voice traffic, among others.

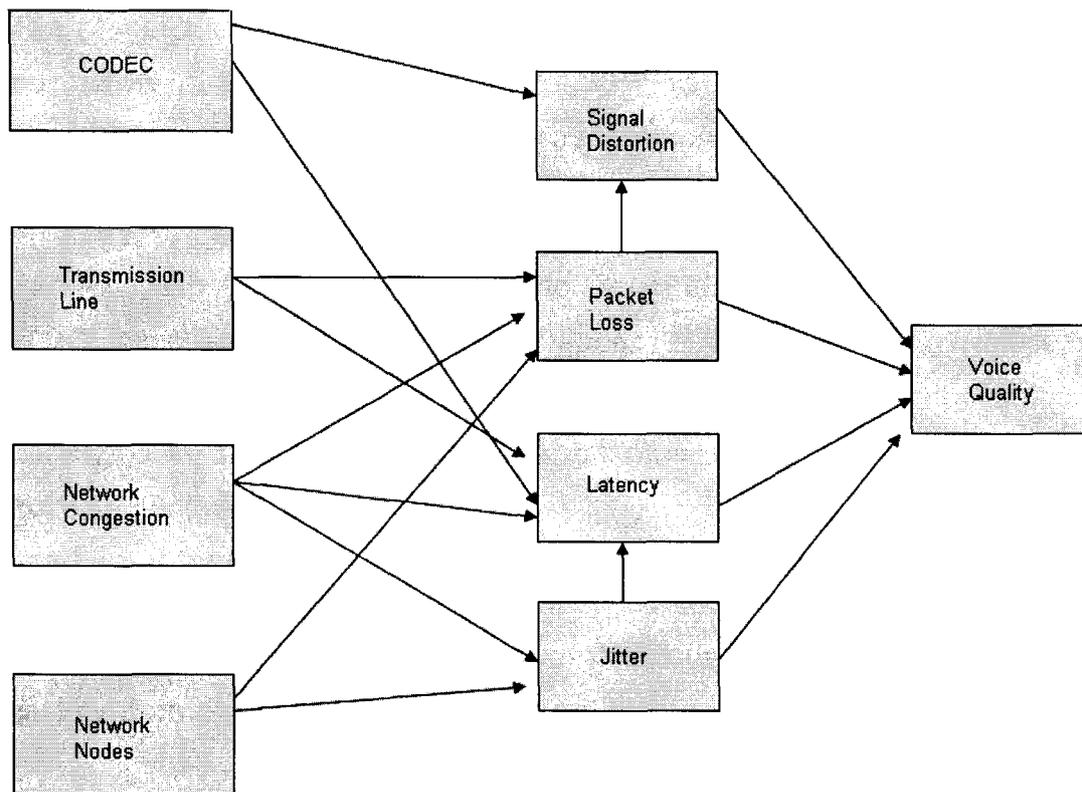


Figure 3.1: VoIP degrading quality of service impairments

The incorrect dimensioning of networks, their traffic, the lack of statistical information, the random deviation from the mean traffic and its distribution during the peak hours affect directly the Quality of Service (QoS) of the media carried by the network.

There are essentially two complementary approaches to testing speech quality

-Subjective tests, which seek the average opinion of users.

-Objective tests, which generate an instrumental measure of average user opinion [2].

3.2 Quality assessment strategies

3.2.1. Subjective quality assessment

The prime criterion for the quality of audio and video communications services is subjective quality, the users' perceptions of services quality. This can be measured through *subjective quality assessment*. The most widely used metric is the mean opinion score (MOS) [30].

The following Absolute Category Ratio (ACR) test is the most frequently used in ITU-T: excellent (5), good (4), fair (3), poor (2), and bad (1). Equivalent wording should be used in languages other than English, which might result in small variations in the original score. The arithmetic mean of all the opinion scores collected is the MOS.

However, while subjective quality assessment is the most reliable method, it is also time-consuming and expensive. Methods for estimating the subjective quality from physical measurements are thus desirable. In order to achieve this, physical quality parameters must be defined.

3. 2. 2. Objective quality assessment

Objective quality assessment methodologies that exploit network and terminal quality parameters to produce estimates of conversational MOS are called *opinion models*. On the other hand, those that require speech signals as inputs and produces estimates of listening MOS are called *speech-layer objective models*, and those that exploit IP packet characteristics and produce estimates of listening MOS are called *packet layer objective models*. Although the speech- and packet-layer objective models estimate the same thing (i.e., the listening quality), they are used in different scenarios depending on the viability on capturing packetized or physical speech samples [1]. There are two main classes of measurement: intrusive (active) and non-intrusive (passive).

3.3 Types of measurement techniques

3. 3. 1 Intrusive (active)

Intrusive techniques inject test signals into a system so they can be captured and assessed at a further point, Figure 3.2. Intrusive assessment involves a comparison between the injected and captured signals.

This method:

- enables isolation of the system under test
- its capable of high accuracy
- can be used during development, commissioning , and routine monitoring
- is possible when no customers are on a system (during development before live traffic is present)
- allows control over external factors
- may incur a real cost, e.g. call charges when testing third-party networks

3. 3. 2 Masked-error models

One of the first applications of perceptual models for quality assessment was proposed by Schroeder et al [3], who used a simple masking method to estimate the audibility of coding noise in a speech coder. This was extended by Brandenburg to give a measure of the mean noise to masking ratio (NMR).

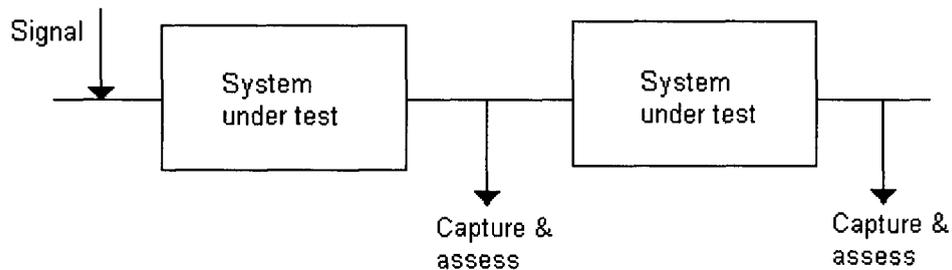


Figure 3.2: Illustration of an intrusive test

These methods basically assume that any (time-domain) difference between the original and processed signals is noise, leading to poor performance when this does not hold, for example in filtering, phase jitter, or re-synthesis.

3. 3. 3 Models based on comparison of auditory transforms

In [4], a more general technique for estimating error audibility based on auditory spectrum distance (ASD), a comparison of audible time-frequency-loudness representations is introduced. This approach can be adapted to simulate a much wider range of perceptual effects, and has been much more successful. Although a successful implementation was demonstrated by Karjalainen, many authors did not cite this work [3].

Similar objective metrics were used in many models, and as recently as 1998 in the measuring normalizing blocks mode (MNB), which used a multi-scale method to compute a quality score from the difference between logarithmic spectrograms of the signals. For intrusive applications, however, the perceptual approach has become dominant [3], [5], [6].

Several new perceptual models for measuring the quality of speech and audio coders emerged in the early 1990s. In [7], an approach similar to that of [4] was taken, but without temporal masking, to compute loudness on a Sone scale in Bark bands and evaluate the mean squared Bark spectral distance (BSD) .

Perceptual audio quality measure (PAQM) introduced the asymmetry factor, weighing the difference in each time-frequency cell by the power ratio of the reference and degraded signals. This amplifies loud additive distortions, emulating perceptual streaming. This was adapted, including the removal of masking, into a method for speech coder evaluation known as the perceptual speech quality measure (PSQM) [5].

3. 3. 4 Models for network testing

Most of the models described above were developed for testing speech or audio coders. Real networks introduce level changes, unknown delay and/or linear filtering – all of which may vary dynamically. If these are not taken into account, they may lead to large false errors being observed in intrusive models that use the method of comparison of auditory transforms, causing highly inaccurate quality scores [3].

Time-delay proved to be a significant challenge as VoIP became widespread in the late 1990s. Variations in packet delay or dynamic jitter buffer re-sizing may lead to a change in the end-to-end audio delay. The comparison method requires the reference and degraded signals to be aligned, but few early models provided an algorithm for delay assessment.

A competition was promoted by the International Telecommunications Standardizations Sector (ITU-T) for solving this problem; a mixture of the methods PSQM and PAMS [8] won and became the new standard ITU-T P. 862 in 2001, the new model is known as Perceptual Evaluation of Speech Quality (PESQ). This method uses the time alignment of PAMS and the auditory transform of PSQM. The average correlation of PESQ with MOS on both known and unknown subjective test data was found to be 0.935 in the ITU-T evaluation [6] and has become one of the most used methods for perceptually evaluating speech quality.

The structure of PESQ is shown in Figure 3.3. The model begins by level aligning both signals to a standard listening level. They are filtered (using an FFT) with an input filter to model a standard telephone handset. The signals are aligned in time and then processed through an auditory transform similar to that of PSQM. The transformation also involves equalizing for linear filtering in the system and for gain variation. Two distortion parameters are extracted from the disturbance (the difference between the transforms of the signals), and are aggregated in

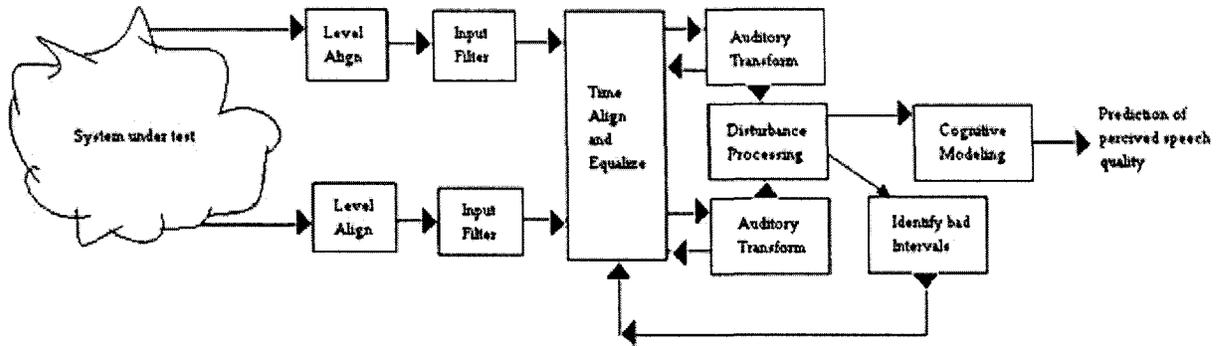


Figure 3.3: ITU – T P. 862, Perceptual Evaluation of Speech Quality (PESQ) block diagram.

frequency and time and mapped to a prediction of subjective mean opinion score (MOS) [9].

Tables 2.1 and 2.2 show the correlation and residual error distribution for PESQ for 38 subjective tests those were available to the developers of PESQ. These included a wide range of simulated and real network measurements. All of this data relates to subjective listening tests carried out on the absolute category rating (ACR) listening quality (LQ) opinion scale. Test material consists of natural speech recordings of 8-12s in duration, with four talkers (two male, two female) for each condition. The results are calculated per condition unless otherwise stated [9].

Table 3.1: Correlation coefficient, 8 unknown subjective tests (PESQ)

Test	Type	Corr.
1	Mobile; real network measurements	0.979
2	Mobile; simulations	0.943
3	Mobile; real networks, per file only	0.927
4	Fixed; simulations, 4-32 kbit/s codecs	0.992
5	Fixed; simulations, 4-32 kbit/s codecs	0.974
6	VoIP; simulations	0.971
7	Multiple network types; simulations	0.881
8	VoIP frame erasure concealment; simulations	0.785

Table 3.2: Error distribution, 7 unknown subjective tests (PESQ). Test 3 excluded as data was per-file only.

Absolute error range	<0.25	<0.50	<0.75	<1.0	<1.25
% Errors in range, PESQ	72.3	91.1	97.8	100	100

3.4 Non-intrusive (passive)

Non-intrusive techniques monitor live network traffic to determine the perceived quality; non-intrusive techniques allow wider scale testing at a reduced operational cost compared to intrusive monitoring. However, these measurement techniques are more difficult to develop and are typically less accurate than intrusive techniques. Substantial advances have been made in the development of non-intrusive techniques for general network applications and for packet-networks [2].

3.4.1 Passive method 1

In contrast to intrusive methods of quality measurement that are based on the transmission of test VoIP calls, the proposed passive method in [10] does not require the transmission of any reference speech signal. Instead, the passive method operates directly on VoIP packet streams that are copied from the network at, for example, an Ethernet hub, a tap on a transmission link, or a Switched Port Analyzer (SPAN) port configured on a switch or router. The passive method also applies the ITU-T objective speech quality measurement algorithms in a novel manner by using them to compare the copied voice signals with 'pseudo packet' stream signals that are constructed in a special way to be described.

As shown in Fig. 3.4, a known voice test signal $f(t)$ is encoded and packetized using the encoding and packetization method that has been identified previously for the copied call. The RTP packet payloads in the copied call are then replaced by the RTP payloads in the constructed packetized voice test signal, as follows.

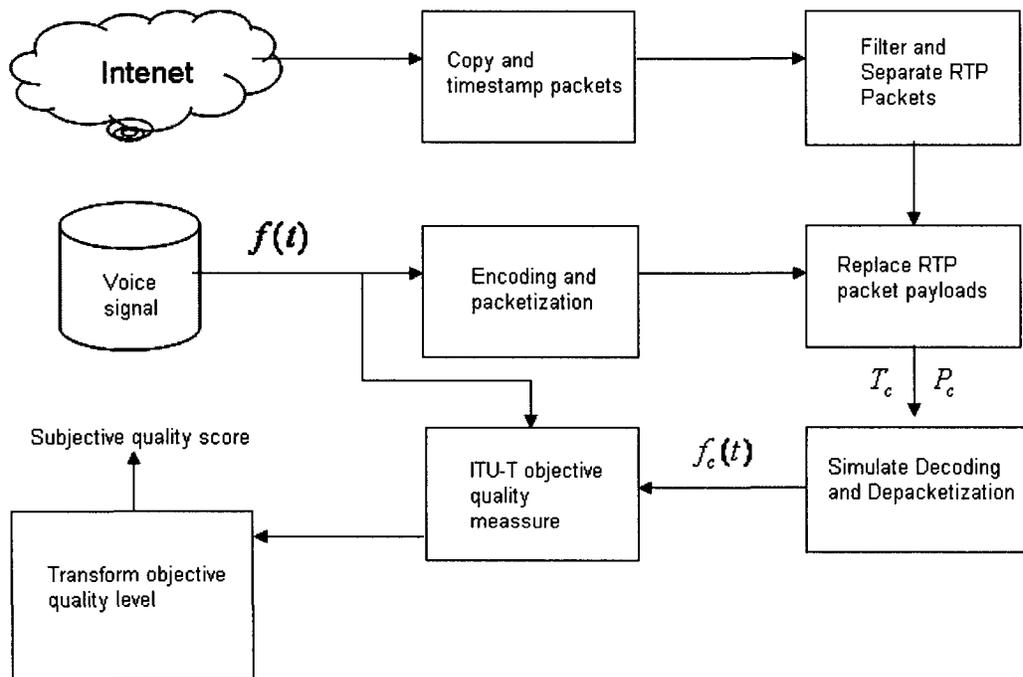


Figure 3.4: Passive method for objectively measuring the subjective voice quality of live VoIP calls.

Let the copied RTP packets for call c be enumerated from 1 to N_c , where N_c is the total number of packets copied for call c . Let $Sc(x)$ be the RTP sequence number found in packet x in call c . Define

$$\begin{aligned} Minc &= \text{Min}\{Sc(1), \dots, Sc(N_c)\} \\ Maxc &= \text{Max}\{Sc(1), \dots, Sc(N_c)\} \end{aligned} \quad (3.1)$$

Also, let the RTP packets of the packetized test signal be enumerated from 1 to N_{test} , where

$$N_{test} \geq Maxc - Minc + 1 \quad (3.2)$$

The procedure is then to replace the payload of packet x in call c by the payload of packet y in the packetized test signal, where

$$y = Sc(x) - Minc + 1 \quad (3.3)$$

In this way, one takes into account the missing and reordered sequence numbers in the copied packets of call c . The resulting modified packet stream for call c may be termed a '*pseudo-packet*' stream.

Having constructed the RTP pseudo-packet stream for call c , the next step is to obtain the decoded signal $fc(t)$ for the pseudo-packet stream. Let $Tc(x)$ denote the time-stamp associated with packet x in pseudo-call c and let

$$Tc = \{Tc(1), \dots, Tc(N_c)\} \quad (3.4)$$

Also, let Pc denote the set of RTP packets in pseudo-call c . The decoded signal $fc(t)$ for pseudo-call c is obtained by simulating the depacketization and decoding method identified previously for call c . In the simulation, the set Tc is used as a set of arrival times for the set of RTP packets Pc .

The remaining step is to determine the quality level for the decoded signal $f_c(t)$. This is obtained by applying an existing ITU-T objective measurement method, such as PESQ, to $f_c(t)$ with $f(t)$ as the original reference signal.

3. 4. 2 Passive method 2

In [12], the authors present a methodology for measurement based on instrumental QoS evaluation of VoIP which allows an efficient comparison for different packet loss situations (e.g., different burst cases), and additionally facilitates time-varying speech quality measurement on a perceptual basis. The main idea is to start with a packet trace based on measurements, in a Testbed, and chose arbitrary trace fragments whose length corresponds exactly to the speech sample under test. It allows an efficient comparison for different packet loss situations (e.g., different burst cases), and additionally facilitates time-varying speech quality measurement on a perceptual basis.

The main contributions of the method are two: they introduce a new instrumental evaluation method for time-varying quality based on applying overlapping trace

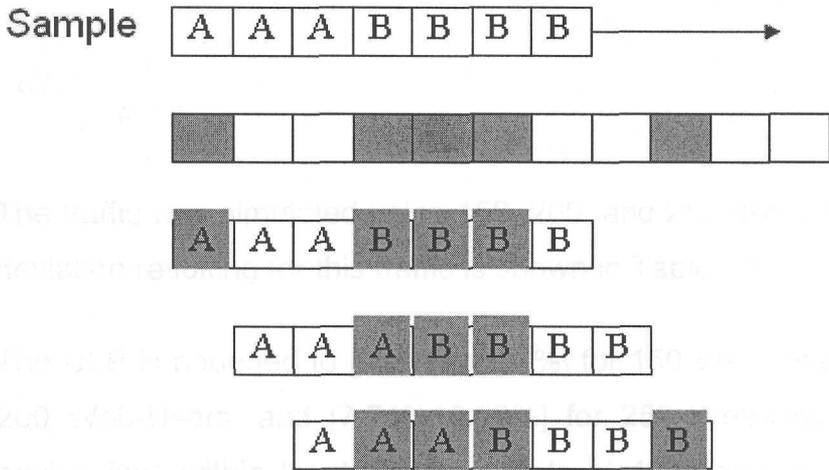


Figure 3.5: Sliding of a speech sample along a packet trace. “A” and “B” denote important and unimportant packets, respectively.

fragments on a suitable speech sample. Using that approach it is demonstrated that even for fixed packet loss rates, VoIP quality may vary significantly.

Figure 3.5 shows how a speech sample, containing important packets (“A”) and less important packets (“B”), is sliding along a packet trace, with dark fields within the trace, indicating individual lost packets.

For testing the degradation of voice quality depending on packet losses, traffic is simulated for client-server web pages petition. The goal is to relate the total packet loss rate and the amount of users generating the traffic, represented with a four state Markov model as follows: p is the probability that a packet will be dropped given that the previous packet has been received, q is the probability that a packet will be received given that the previous packet has been lost. $1-q$ is termed the “conditional loss probability” (CLP), which serves as an indicator for the loss burstiness of the traffic:

$$CLP = 1 - q \tag{3.5}$$

The unconditional loss probability (ULP) represents the average packet loss rate and can be calculated as:

$$ULP = \frac{p}{p + q} \tag{3.6}$$

The traffic was simulated using 150, 200, and 250 users, the mean and standard deviation resulting for this traffic is shown in Table 3.3.

The ULP is bounded to [0.00%; 2.87%] for 150 Web-Users, [0.87%; 10.49%] for 200 Web-Users, and [7.74%19.60%] for 250 Web-Users. Thus, the range of packet loss within the trace fragments of 400 packets overlap to a significant extent.

Table 3.3: Means and standard deviations for the ULP for different number of web-users.

No users	ULP %	
	mean	stddev
150	0.89	0.56
200	5.26	1.50
250	13.77	1.94

3.5 Methods for network testing (Analysis of network performance)

The subjective quality of a received voice signal depends essentially on network-level QoS only through the accumulated loss, jitter, and reordering effects at the packet level. Consequently, if one copies a VoIP packet stream from a point in the network while time-stamping the copied packets so as to be able to later regenerate their temporal sequence, then the quality of the received voice signal can, in principle, be determined by comparing the depacketized and decoded signal with the original transmitted signal [10].

The subjective quality of a received voice signal depends essentially on network-level QoS only through the accumulated loss, jitter, and reordering effects at the packet level. Consequently, if one copies a VoIP packet stream from a point in the network while time-stamping the copied packets so as to be able to later regenerate their temporal sequence, then the quality of the received voice signal can, in principle, be determined by comparing the depacketized and decoded signal with the original transmitted signal [10].

The strong dependence of the subjective quality of a voice signal and the state of the network raises the necessity to explore the correlation among the network parameters and the distortion of the physical signal. The test beds used at [11-12] identify the most influent network parameters affecting the quality of voice as the bandwidth of the bottleneck link, size of the bottleneck buffer, propagation delay (D), and the average of data size transmitted, see Figure 3.6.

The experiments in [11] are set to study 6 cases. Case 1 serves as reference. The other cases are used to examine the influence of network parameters. Case 2 and case 3 are for the size of bottleneck buffer, cases 4 and 5 are for the propagation delay, and case 6 is for the transmitted file size. Each case is subdivided in 4 cases to examine the bottleneck of link capacity. In each experiment, 240 voice signals of 10.56 sec are transmitted with a period of 16 sec.

Note that PESQ does not evaluate the mean delay of a VoIP signal. The mean delay is also an important factor for voice quality and should be less than 150 msec for adequate performance [11].

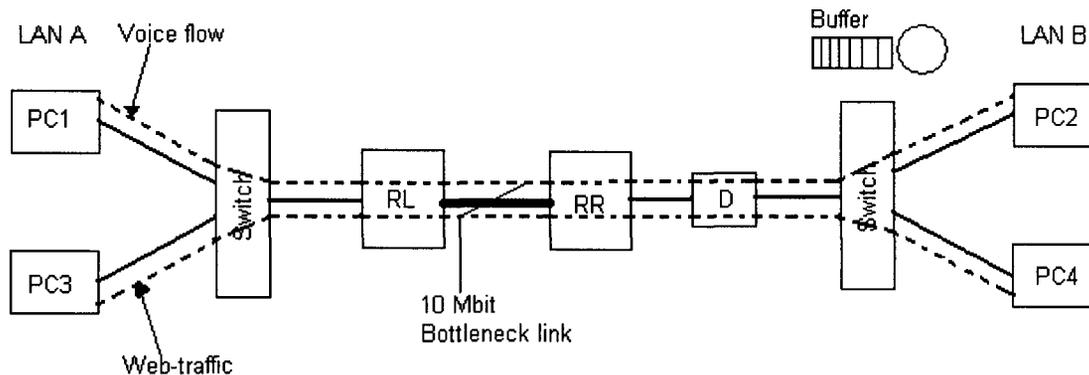


Figure 3.6: Testbed.

Analysis of network performance

Every time the voice quality is calculated, the network performance is also calculated. The following statistical factors are used in [11] for this purpose: average of jitters (J_{av}), standard deviation (J_{sd}), standard deviation of packets interarrival times (I_{sd}), and packet loss ratio (R_{loss}). Let the arrival times of the k^{th} VoIP packet observed in LAN-A and LAN-B during a period of 16 sec be T_k^A and T_k^B ($k = 1, 2, \dots, K$), respectively. The values are calculated as:

$$J_k = T_k^B - T_k^A - D \quad (3.7)$$

$$J_{av} = \sum_{k=1}^K J_k / K \quad (3.8)$$

$$J_{sd} = \sqrt{\sum_{k=1}^K (J_k - J_{av})^2 / (K - 1)} \quad (3.9)$$

$$I_k = T_{k+1}^B - T_k^B \quad (3.10)$$

$$I_{av} = \sum_{k=1}^K I_k / K \quad (3.11)$$

$$I_{sd} = \sqrt{\sum_{k=1}^K (I_k - I_{av})^2 / (K - 1)} \quad (3.12)$$

$$R_{loss} = N_{loss} / N \quad (3.13)$$

$$N = K + N_{loss} \quad (3.14)$$

All the correlation coefficients in [11] between each of the network performances defined in Cases above and the voice quality (PESQ score) for every sub-case are calculated. From which the following outcomes:

All the network performances show a certain level of correlation with voice quality. In particular some values of correlation coefficients of J_{av} , U, and I_{sd} averaged for all sub-cases are larger than 0.7. In comparison R_{loss} present a weaker correlation compared with others.

Correlations of each network performance except for J_{sd} seem insensitive to the changes in network conditions, the coefficients are close to each other. On the contrary, the correlation of I_{sd} depends on the bottleneck bandwidth.

From the scattering plots presented at [11] it's conclude that plots of the same channel capacity are similar to each other, which means that the correlation features depend on C but rather insensitive to Delay, Buffer Size, and the Pareto distribution parameter Δ used to generate random packet sizes for tests.

This chapter makes a compilation of some of the useful, standardized and commercially applied models and methods for the Voice over IP quality of service. The techniques described here were of 2 kinds, speech layer objective measurements and packet layer objective measurements. The speech layer objective measurements is not the criterion for the analysis developed in this work, but rather it is a point of reference to support the results of the proposed methodology in this job.

CHAPTER IV NETWORK MEASUREMENTS AND IMPLEMENTATION OF VOIP MONITOR

Capturing RTP packets containing a media stream through a given network allows tracking the evolution of the type of media traffic and bandwidth used. It is possible to infer from network objective statistical parameters, (e.g., latency, jitter, and packet loss), the mean quality perceived by the users. Monitoring in real time provides accurate information about the distribution of the impairments degrading the voice quality.

Some advantages of designing a VoIP sampler are:

1. Performance Monitoring

Traffic studies help benchmark existing circuits to determine call concurrency, capacity and call flow patterns to establish bandwidth requirements. Once IP telephony is implemented, voice quality is measured to ensure standards are met.

2. Monitor quality of service for the converged network: Real-time Operation and Reporting Capabilities:

Provides around-the-clock monitoring of voice quality and the impact of voice on critical business hours, the network manager can schedule network resources or opt for increasing its capabilities to ensure a level of voice quality.

A disadvantage about sampling a VoIP streams and its statistical network parameters are: the cost of the sample time, resources used, and the perturbation of the network traffic reducing the available bandwidth.

4.1. Issues to consider when designing a VoIP QoS monitor

In [13], the Internet Engineering Task Force (IETF) working group for Internet Performance Metrics (IPMM) provides some major issues to consider when designing a sampler for Network performance measurement with non periodic streams. The ones fitting our necessities are:

- Measurements of many network impairments (e.g., delay variation, consecutive loss, reordering) are sensitive to the sampling frequency. When the impairments themselves are time-varying and the variations are somewhat rare, yet important, a constant sampling frequency simplifies analysis.
- The total traffic generated by this or any sampling method should be limited to avoid adverse affects on non-test traffic (packet size, packet rate, and sample duration and frequency should all be considered).

4.2. The configuration parameters of periodic sampling

- T , the beginning of a time interval where a periodic sample is desired.
- dT , the duration of the time interval between sample start times: Good sample start times for our proposed monitor can be specified to any value above 8 seconds, the sample duration.

Global metric parameters

- Src , the IP address from where the packetized voice stream originates.
- Dst , the IP address where the samples are taken.
- $dTloss$, the maximum waiting time for a packet before declaring it lost.
- Packet Type, Real-Time Transport Protocol (RTP).
- $Tcons$, a safety time interval for allowing the calculation of estimates from collected data at the measurement points.

The packet size is constant during the total sampling time; it can be changed at the beginning of a new simulation if a different fragment of audio is to be used.

4.2.1 Parameters collected at the measurement point MP (Source LAN)

- Tstamp(i), for each packet (i), the departing time of the packet as measured at MP(Src).
- PktID(i), for each packet (i), a unique identification or sequence number.

4.2.2 Parameters collected at the measurement point MP (Destination LAN)

- Tstamp(i), for each packet (i), the arrival time of the packet as measured at MP(Dst).
- PktID(i), for each packet (i), a unique identification or sequence number.
- dTstop, a time interval, used to add to time Tf to determine when to stop collecting metrics for a sample.
- PktStatus (i), for each packet (i), the status of the packet received. Possible status includes OK, packet header corrupt, packet payload corrupt, duplicate or fragmented.

4.3. Metrics to sample

4.3.1 One-way-Delay Metric [14]

For a real number dT , the One-way-Delay from Src to Dst at T is dT , means that Src sent the first bit of an RTP packet to Dst at wire-time T and that Dst received the last bit of that packet at wire-time $T+dT$. It does not make any sense to report a negative delay value.

The minimum value of this metric provides an indication of the delay due only to propagation and transmission delay. It also suggests when the path is lightly loaded just comparing with a reference point like 200 ms, which means that values above 200 ms represent a congested network. This metric is modeled as:

$$D(i) = (R_i - R_{i-1}) - (S_i - S_{i-1}) = (R_i - S_i) - (R_{i-1} - S_{i-1}) \dots\dots\dots (4.1)$$

$$= DestinationArrivalTime(i) - SourceArrivalTime(i-1) - T$$

Where:

D_i = One-way-Delay for packet i

R_i = Time stamp at Destination for packet i

R_{i-1} = Time stamp at Destination for packet i-1

S_i = Time stamp at Source for packet i

S_{i-1} = Time stamp at Source for packet i-1

T = Size of voice sample in each packet (e.g., 20 ms for codec. 711)

Packet (i) and Packet (i-1) stands for packets received successfully and played after being queued in a de-jitter buffer.

In quality-of-service (QoS) enabled networks, provisioning in one direction may be radically different than provisioning in the reverse direction, and thus the QoS guarantees differ. Measuring the paths independently allows the verification of both guarantees.

The One-way-Delay from Src to Dst at T is undefined (informally, infinite), means that Src sent the first bit of a Type-P packet to Dst at wire-time T and that Dst did not receive that packet.

Synchronization of source and destiny is needed for accurate measurements of this metric, the Network time protocol (NTP) [30] has precision of several milliseconds what makes it suitable for our design proposals.

If a packet is duplicated only the first-arrival packet with the sequence number of the duplicated packet must be considered.

4. 3. 2 IP Packet Delay Variation Metric or Jitter (ipdv) [15]

One important use of delay variation, also known as jitter, is the sizing of play-out buffers for applications requiring the regular delivery of packets (for example, voice or video play-out). What is normally important in this case is the maximum One-way-Delay variation, which is used to size play-out buffers for such applications. This metric is modeled as in [18], which is a first order estimator better suitable for VoIP delay variation:

$$\begin{aligned} J(i) &= J(i-1) + (|D(i-1,i) - J(i-1)|) / 16 \\ &= J(i-1) + (|D(i) - D(i-1) - J(i-1)|) / 16 \end{aligned} \quad (2)$$

Where:

$J(i)$ = Jitter for packet i

$J(i-1)$ = Jitter for packet i-1

$D(i)$ = One-way-Delay for packet i

$D(i-1)$ = One-way-Delay for packet i-1

In addition, this type of metric is particularly robust with respect to differences and variations of the clocks of the two hosts. This allows the use of the metric even if the two hosts that support the measurement points are not synchronized.

Example:

$$\text{Jitter} = ((T_{\text{destiny}}(i) + \text{imprecision at destiny}) - (T_{\text{source}}(i) + \text{imprecision at source})) - [((T_{\text{destiny}}(i-1) + \text{imprecision at destiny}) - (T_{\text{source}}(i-1) + \text{imprecision at source}))]$$

Jitter is defined for two packets from Src to Dst, as the difference between the value of the delay from Src to Dst at T2 and the value of the One-Way-Delay from Src to Dst at T1. T1 is the wire-time at which Src sent the first bit of the first packet, and T2 is the wire-time at which Src sent the first bit of the second packet. This metric is derived from the One-way-Delay metric.

Positive Jitter means that packet sent at wire time T1 arrived at destiny before packet sent at wire time T2. Negative Jitter means that second packet arrived before the first packet.

Experiments have shown that Jitter is the most negative effect degrading the voice quality.

4. 3. 3 One-way Packet Loss Metric [16]

Thus, One-way-Packet-Loss is 0 exactly when One-way-Delay is a finite value, and it is 1 exactly when One-way-Delay is undefined.

The One-way-Packet-Loss from Src to Dst at T is 0, means that Src sent the first bit of a packet to Dst at wire-time T and that Dst received that packet.

The One-way-Packet-Loss from Src to Dst at T is 1, means that Src sent the first bit of a type-P packet to Dst at wire-time T and that Dst did not receive that packet.

If a packet is corrupted impeding to clearly read the source, destination, sequence number or content, then it is dropped. If multiple copies of a packet are received through different paths, the first of the copies is considered as the received packet and the remaining copies are dropped.

The measurement instruments should be calibrated such that the loss threshold is reasonable for application of the metrics and the clocks are synchronized enough so the loss threshold remains reasonable.

4.4. RTP: A Transport Protocol for Real-Time Applications [19]

RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The

data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide minimal control and identification functionality.

The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence, but sequence numbers might also be used to determine the proper location of a packet, for example in video decoding, without necessarily decoding packets in sequence.

The Internet, like other packet networks, occasionally loses and reorders packets and delays them by variable amounts of time. To cope with these impairments, the RTP header contains timing information and a sequence number that allow the receivers to reconstruct the timing produced by the source. To estimate how many packets are lost, the receiver uses the sequence number.

Synchronization source (SSRC): The source of an RTP packet stream, identified by a 32-bit numeric SSRC identifier carried in the RTP header so as not to be dependent upon the network address. All packets from a synchronization source form part of the same timing and sequence number space, so a receiver groups packets by synchronization source for playback.

If the samples are intended to be taken for a tripartite VoIP call, there will be 3 SSRC involved, they can then be used to identify the source of each of the three packetized voice streams.

Figure 4.1 shows the parameters composing the RTP header, the ones used by our sampler are:

Sequence number: 16 bits

The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number should be random to make known-plaintext attacks on encryption more difficult.

Timestamp: 32 bits

The timestamp reflects the sampling instant of the first octet in the RTP data packet. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent.

For converting the RTP timestamp from samples to time, it is divided by $2^{32} - 1$.

If a tripartite VoIP call is to be sampled, each source begins with a different random time stamp.

All the participants in the conversation to be sampled must have a reference clock (e.g., NTP), this is fundamental to measure timestamps accurately.

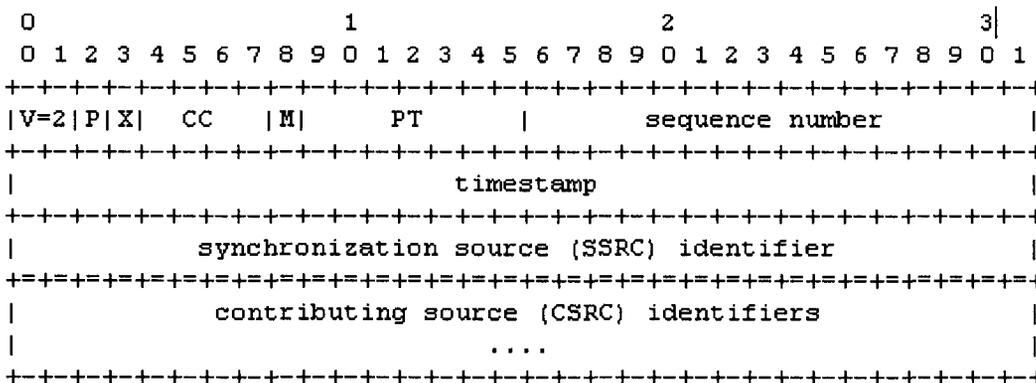


Figure 4.1: RTP Fixed Header Fields

4.5. Implementing a VoIP QoS Monitor

The concepts numbered through this chapter provide us a map for programming a VoIP QoS Monitor as depicted in Figure 4.2. Now we propose a method and some tools to materialize these concepts. It is assumed that the packetized voice stream includes for each packet an RTP header.

There are 3 tools needed to create a real time packet monitor, these are:

1. Wireshark is a packet analyzer able to capture packets specifying the source and destination for VoIP communication. This packet sniffer has the characteristic of being able to capture any kind of traffic desired, our case is RTP/UDP/IP packets, and one of the most important desired characteristics is the viability to operate from command line for easy integration in a programming language that allows execution of external command-line applications. A command for wireshark to capture RTP media stream from a given SLAN to DLAN is:

```
!"C:\Program Files\Ethereal\tethereal" -a "duration:8.2" -i "2" -t "a" -R
"ip.src==192.168.0.75 && ip.dst==192.168.0.1 && rtp" -T "text" -V>>test1.txt
```

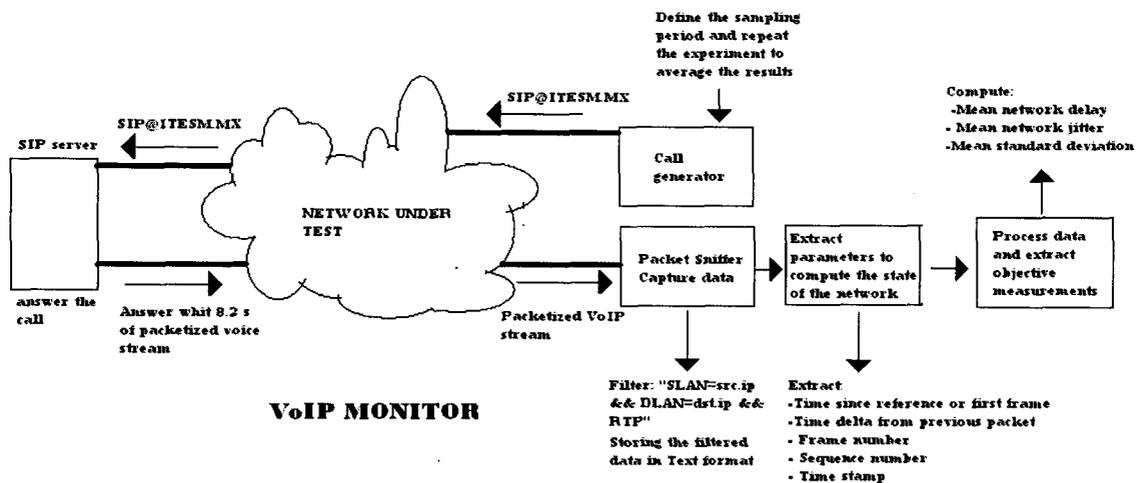


Figure 4.2: VoIP QoS Monitor.

where:

Thetereal launches the sniffer program and starts capturing.
 Duration:8.2, is the total length of the capture (e.g., 8.2 s).
 -i "2", specifies the capture interface (e.g., LAN card).

-R "ip.src==192.168.0.75 && ip.dst==192.168.0.1 && rtp" is a filter that selects among the captured packets type, RTP, and the packetized media data transmitted from SLAN=ip.src to DLAN=ip.dst.

-T "text", instruction to save and append the captured packets in Text files.

2. Express Talk is a Softphone with the advantage of being controllable from command line, you can install the most basic Express Talk version, which is downloadable free. An example to control Express talk from command line is:

```
!"C:\Program Files\NCH Swift Sound\Talk\talk.exe" -dial "206@192.168.0.75"
```

where:

Talk.exe: Launches Express Talk and registers it at the user SIP registrar server.

-dial 206@192.168.0.75: specifies the username and domain of the destination user.

3. Matlab is used in combination with the tools mentioned in the last two points to process the data extracted from each RTP/UDP/IP. All the packets captured are stored by wireshark in text format.

Matlab has all the necessary tools needed to process the data stored in text file and compute the statistics proposed in [13-14], and equation (3.12).

The ideal solution should work with any equipment within your voice and data networks, ensuring its usefulness as the infrastructure grows.

CHAPTER V RESULTS

In this chapter, a method for evaluating signal degradation as a function of packet delay statistics is developed for VoIP networks. The method involves the representation of a signal as discrete tones at speech frequencies. The signal is then separated in 20-ms packets and each packet is randomly delayed according to different probability distributions. This means, the signal is treated as an audio packetized stream sent through an IP Network with random delays. The analysis then is based on the autocorrelation and power spectral density of the resulting signal.

The effect of network delay is accounted in the model by two ways. First, the signal is analyzed assuming that each packet never arrives before its predecessor by traversing the path from source LAN (SLAN) to destination LAN (DLAN). In other words, packets arrive in chronological order. This allows the study of signal degradation due only to a random-delay linear discrete filter. The departure time between packets is assumed to be 20 ms corresponding to the packet length in time. This assumption is based on the fact that modern processors where VoIP applications run have a negligible coding delay ($\sim 100 \mu\text{s}$). Indeed, even in machines where coding delay is important, the model works because it takes into account a single delay that represents (see Figure 5.1) the sum of the coding, transmission, queuing, and propagation delay. To exemplify the proposed method, the packet inter arrival time is modeled in two ways: with a Gaussian and with an Exponential random variable for a mean range of 1×10^{-7} to 500×10^{-3} seconds and variances in the same range. The exponential choice is justified because a worst-case scenario corresponds to a Poisson arrival process of the packets, whereas the Gaussian distribution is justified because the

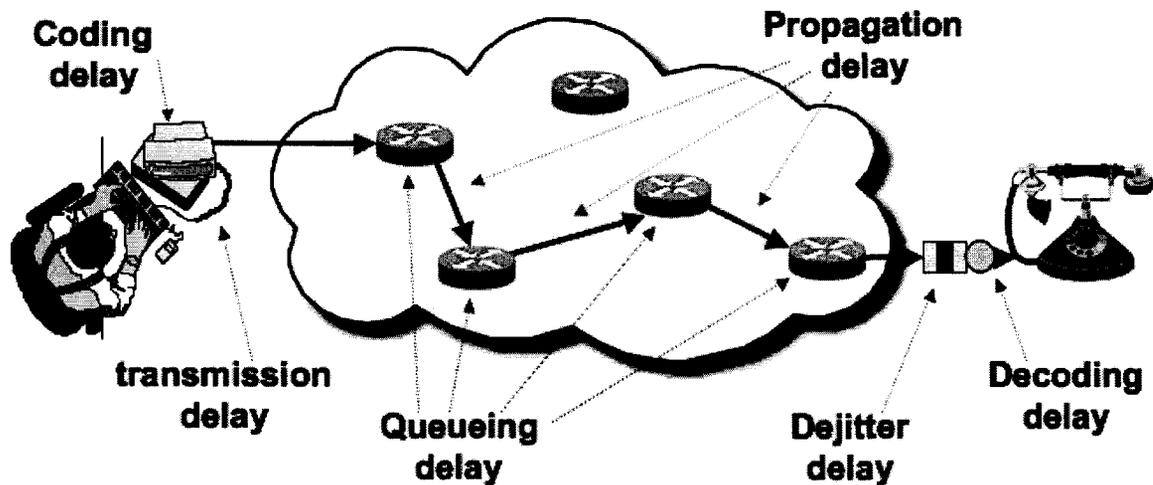


Figure 5.1: Total Latency contributors

total delay is the result of a sum of several random delays. In practice, the Gaussian distribution allows the independent manipulation of first and second-order statistics contrary to the exponential case where specifying the mean simultaneous fixes the variance.

The second way for modeling signal degradation involves the inclusion of a de-jitter buffer waiting for the speech packetized voice stream at the destination side. The buffer size is simulated in the range $100ms - 400ms$ and a new buffer parameter is defined and denominated Superior Quality Threshold (SQT). The SQT is used to control the amount of time the buffer can wait to get full so the person at destination doesn't perceive the lack of audio for very short amounts of time. It is shown that SQT should be below 200 ms.

Results presented for both cases describe the impairments caused by a random delay distribution and the resulting statistics introduced by the loss of packets incurred by excess delay traversing the path from source to destination. Data networks in theory are designed for operation with a BER below 10^{-9} errors per second so this effect is not considered when obtaining the power penalties and phase noise degrading the signal given that 8.2 seconds, the recommended

duration of an audio test signal, is coded in only 524 kbits. This means that the average number of error bits is extremely low ($\sim 10^{-4}$).

Among the results obtained from simulations, the tone spectral linewidth resulting from phase noise induced by random packet delays, the Mean Delay between packets Played (MDP), the Mean Jitter resulting after Playtime (MJP), Mean number of Lost Packets (MLP) and the Mean Standard Deviation after playtime (MSTDP), for both distributions are included and commented. The concepts of autocorrelation and Power Spectral Density are introduced to compute the results presented.

Other effects like packet collisions, retransmitted, and duplicated packets are assumed to be absorbed by the total latency of traversing the path from SLAN to DLAN.

All the results presented assumes the Mean Delay (MD) of packets from SLAN to DLAN to be cycle-stationary, so high peak hours can be modeled by a higher mean delay and variances, with the possibility of monitoring and analyzing in real time network realistic scenarios.

5.1 Ideal scenario for packetized discrete tones stream

Figure 5.2 depicts the case used as first approximation for modeling a packetized tone length of 8.2 seconds as specified in [1]. The 8.2 s are fragmented in 20 ms signals so each fragment is codified and packetized in format IP/UDP/RTP and sent from SLAN to DLAN. The information carried by the IP and RTP headers is used to compute the impairments characterizing the QoS of the VoIP Service in the given Network.

Equation 5.1 is the starting point to describe the impairment affecting the quality of the transmitted signal. τ_i is a Normal or exponential R.V. with mean and standard deviation in the range; $0 < \mu_i < 500$ ms and $0 < \sigma_i < 500$ ms for both cases. It is important to notice that the mean and standard deviation cannot be

varied independently in the case of the exponential inter-arrival packet delay. However, the standard deviation for the Gaussian case can be varied independently from the mean.

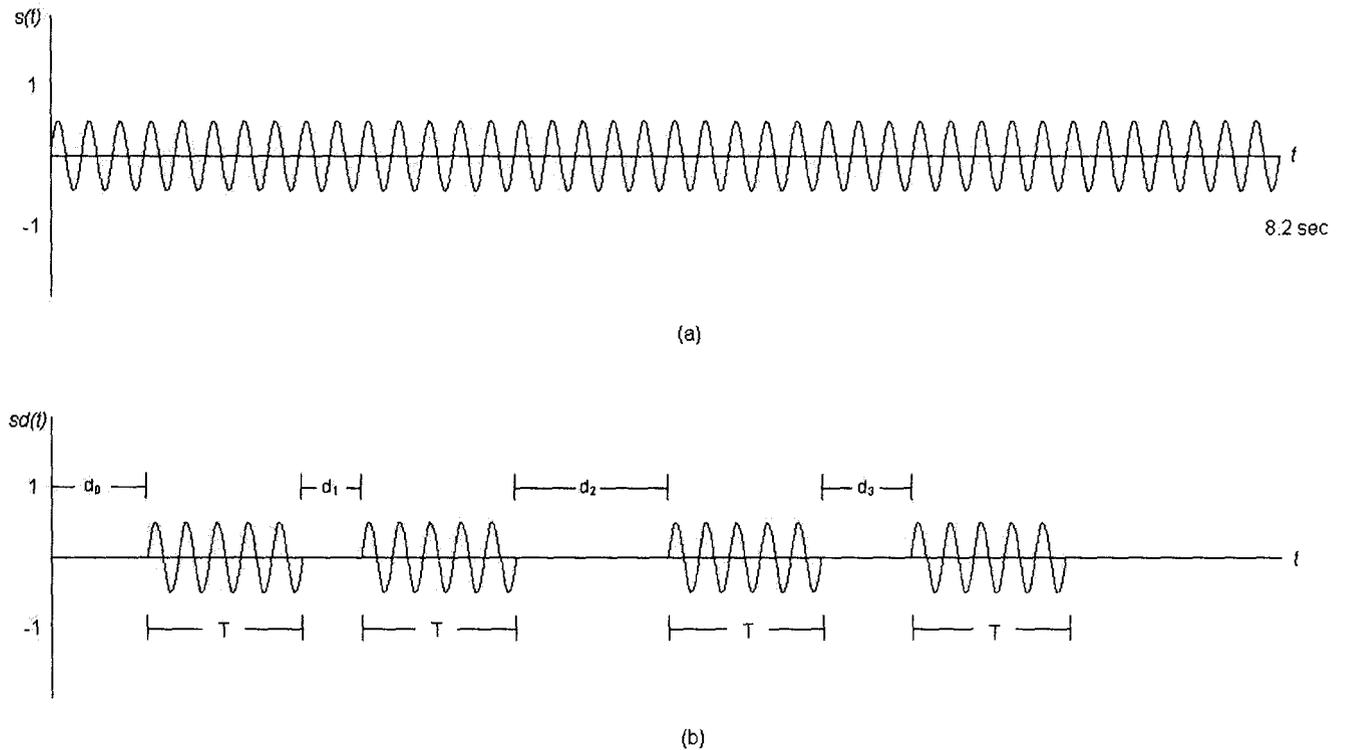


Figure 5.2: 8.2 seconds frequency tone, f_c (a) Original signal and, (b) Packetized signal.

The signal representing the random delayed packetized tone can be represented as:

$$s(t) = \sum_{i=0}^{k-1} \text{Cos}(2\pi f_c(t - \tau_i + iT)) \text{rect}\left(\frac{t - \tau_i - iT - T/2}{T}\right) \quad (5.1)$$

T represents the length of codified voice media carried by each packet, our case assumed to be 20 ms; the rect function is used to window the 20 ms fragments to

represent the packetized media stream from the original speech signal. The R.V. τ_i adds a delay between packets carrying the media stream independently with the following restrictions assumed:

$$\begin{aligned}\tau_0 &= d_0 \\ \tau_1 &= \tau_0 + d_1 = d_0 + d_1 \\ &\vdots \\ \tau_k &= \tau_{k-1} + d_k = d_0 + d_1 + \dots + d_k\end{aligned}\quad (5.2)$$

The addition of k-1 packets re-builds the original voice stream now packetized and delayed by the sum of dependent R.V τ_i . As mentioned earlier, this scenario does not consider the reordering or the lost packets. Next, the correlation of the signal is computed to account for power of the signal using the Wiener-Khintchine theorem. The autocorrelation of the signal in (5.1) is written as:

$$\begin{aligned}R_i^j(h) &= \langle s(t)s^*(t+h) \rangle \\ &= \langle \sum_{i=0}^{k-1} \text{Cos}(2\pi f_c(t - \tau_i + iT)) \text{rect}\left(\frac{t - \tau_i - iT - T/2}{T}\right) \sum_{l=0}^{k-1} \text{Cos}(2\pi f_c(t + h - \tau_l - lT - T/2)) \text{rect}\left(\frac{t + h - \tau_l - lT - T/2}{T}\right) \rangle\end{aligned}\quad (5.3)$$

Simplifying the double sum notation of Equation 5.3 by all the possible combinations of indices l and j, the multiplication of rect functions in the original signal and its delayed conjugate complex representation form a new rectangle; this new rectangle is different of 0 if and only if the autocorrelation variable h is related with i and j in range:

$$\tau_{i+r} - \tau_i - (1-r)T < h < \tau_{i+r} - \tau_i + (1+r)T \quad (5.4)$$

$$\text{rect}\left(\frac{t - \tau_i - iT - T/2}{T}\right) \text{rect}\left(\frac{t + h - \tau_l - lT - T/2}{T}\right) = \text{rect}\left(t - \left(\frac{\tau_{i+r} + \tau_i + (2i + r + 1)T - h}{T - |h|}\right)\right) \quad (5.5)$$

Equation 5.5 changes the variable l by its representation of the form l=i+r, where r can take positive or negative values. Equation 5.5 is only valid if independent variable "h" is in the range specified by Equation 5.4. Considering only one

packet transmission or a tone length of 20 ms (e.g., $i=0$), the resulting random signal has an autocorrelation independent of the R.V. τ_i .

5.2 Poisson Process

Poisson points are specified by the following properties:

(i) the number of points $n(t_1, t_2)$ of the points t_i in an interval (t_1, t_2) of length $t = t_2 - t_1$ is a Poisson R.V. with parameter λt .

(ii) If the intervals (t_1, t_2) and (t_3, t_4) are nonoverlapping then the R.V. are independent, see Figure 5.2.

Equation 5.6 is an impulse train marking the positions of playtime for each packetized fragment of the original stream of media.

$$Z(t) = \sum_{i=0}^{k-1} \delta(t - \tau_i - iT) \quad (5.6)$$

Equation 5.7 is the impulse response of the system used to generate (5.1) from a pulse train like (5.6).

$$H(t) = \text{Cos}(2\pi f_c t) \text{rect}\left(\frac{t - T/2}{T}\right) \quad (5.7)$$

$$N(t) = \sum_{i=0}^{k-1} u(t - \tau_i - iT) \quad (5.8)$$

Since the integral of a delta function $\delta(t - \tau_i - iT)$ is a step function $u(t - \tau_i - iT)$ the Poisson process is generated from (5.6) and has the form of equation (5.8), see Figure 5.2.

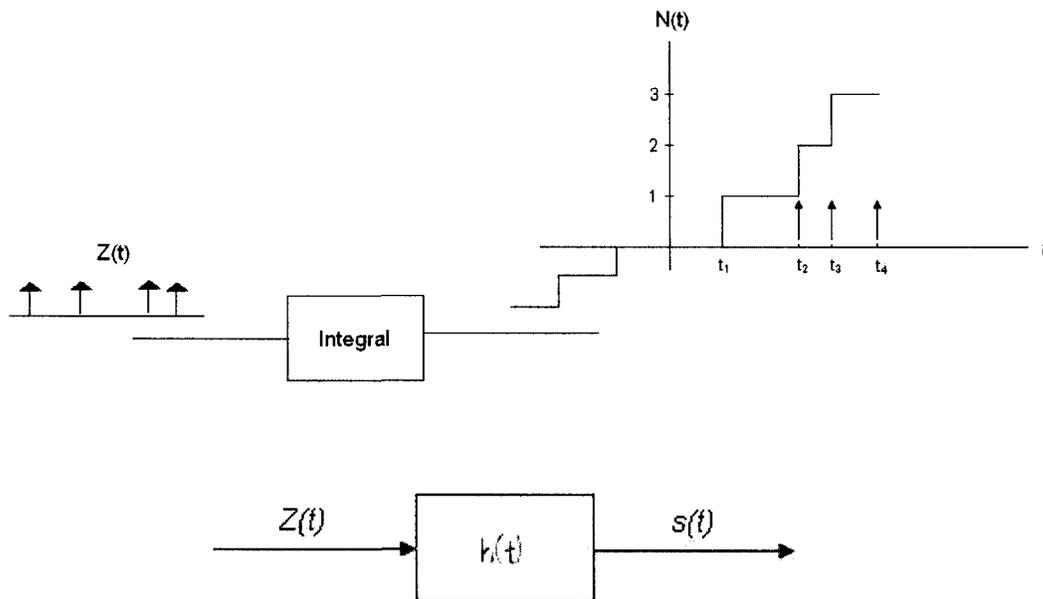


Figure 5.3: System $h(t)$ processing the input $Z(t)$ to generate the output of the form (5.1)

We can generate the signal in (5.1) in the same way a Poisson Process is generated. In Figure 5.3 an impulse train representing the packets interarrival time is processed by the system modeled by equation (5.7) so the signal (5.1) is generated. From this analogy it can be established some similarities between a poisson process and equation 5.1. Basically both has N points distributed along the time but the difference is that process of Equation 5.1 has a rise every time t_i as Poisson process does, see Figure 5.3, but for model (5.1) its pulses maintains up just for T seconds, in our simulations $T=20\text{ms}$ (e.g. G 711). The solution for the autocorrelation of Equation 5.1, which models the behavior of Internet Packets, has not been solved and there is no reference showing these results, it can be proposed as future work to get a close expression for the autocorrelation of equation 5.1.

5.3 De-jittering and Lost Packets Inclusion

A better approximation of the real behavior of a voice packetized stream transmitted from SLAN to DLAN is by representing the noise phase in the form of Equation 5.9, see Figure 5.4., and simulating a de-jittering buffer to account for packet losses, jitter, and delay variation, see Figure 5.5.

$$\begin{aligned}
 A_0 &= \tau_0 \\
 A_1 &= \tau_1 + T \\
 A_2 &= \tau_2 + 2T \\
 &\vdots \\
 A_{k-1} &= \tau_{k-1} + (k-1)T
 \end{aligned} \tag{5.9}$$

Where $\tau_0, \tau_1, \tau_2, \dots, \tau_{k-1}$, are R.Vs instead of the sum of R.Vs as assumed in (5.2), now the packets arrive out of sequence to DLAN, (e.g., Figure 5.4). The natural delay of generating T ms of speech by the source is included by $(k-1)T$.

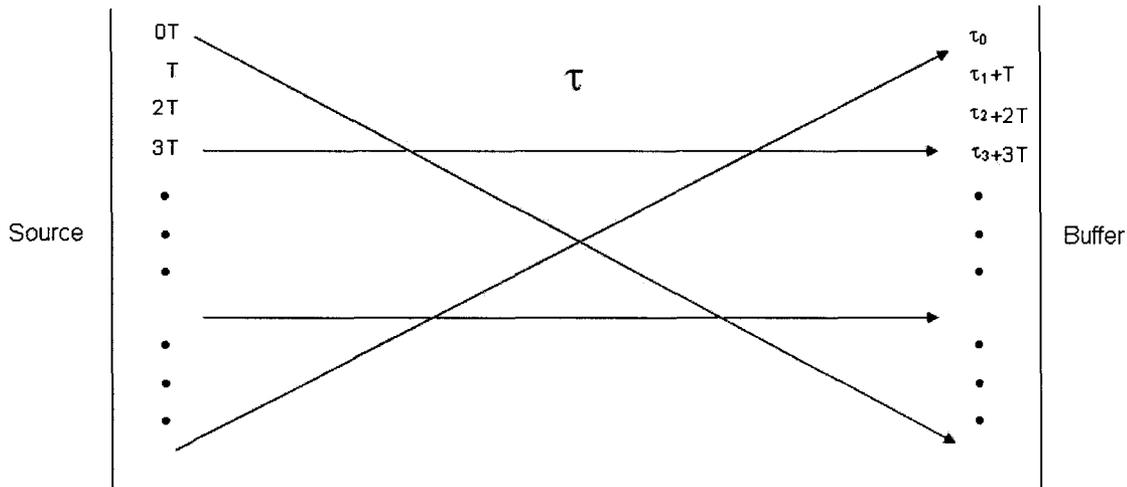


Figure 5.4 A better approach for a real VoIP system communications. See Equation 5.9.

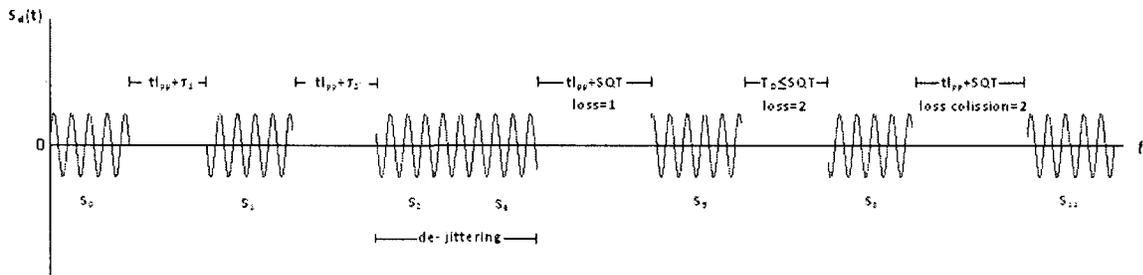


Figure 5.5: Simulation of a De-jitter buffer for an UDP/IP/RTP stream affected by modulation random phase noise.

Equation 5.9 is more realistic in a communication of audio stream, the jitter is now a factor involved in this method. The signal after being processed by a de-jitter buffer should have the shape shown in Figure 5.5.; three cases causing the lost of packets are identified.

The buffer gets full before the packet being played ends.

1. The process of de-jittering is done and the media fragment next to be played is ready before the reproduction of actual fragment of voice ends. When the actual packet being played ends, the next packet better located in sequence is played immediately.
2. The buffer gets full before the end of a maximum fixed delay between reproductions of two packets called Superior Quality Threshold (SQT) ends.

The time of the packet arriving at buffer so the buffer gets full, plus the time of de-jitter the fragments queued, so the next packet to be played is ready before the overflow the SQT, the time reproduction of the packet in sequences resulting from this case is:

$$tl_{pp} < t_{rep} < tl_{pp} + SQT \quad (5.10)$$

Where t_{lpp} is the last packet played time and SQT is the maximum allowable delay for the next packet to be played, below 150 ms is a good limit.

3. The buffer is absolutely empty and the next packet to be played arrives at buffer before the expiration of SQT. No other packet arrives at buffer before SQT ends, the only packet at buffer is played and a new threshold is fixed.

In all the cases above a packet is dropped if the sequence of the arriving packet is lower than the sequence of the last packet played.

A burst lost happens when the buffer gets full and this last packet entering the buffer is forward in sequence than packet not arrived at buffer yet.

There is a condition for the termination of the conversation and happens when the SQT is overflowed and no packet has arrived at buffer, in that moment all the packets left to reach the destination are lost and the call finished. See Figure 5.5.

Equation 5.1 in combination with Equation 5.2 or 5.9, depending the simulated scenario, and the conditions imposed by the de-jitter buffer where used to generate the following results.

5.4 Introducing Correlation and Power Spectral Density for (5.1)

The first simulation was carried out using the approximation from Equation 5.1 and 5.2., an exponential R.V with parameter $1/\lambda$, and a tone centered at 555.3 Hz.

$$0.1\mu s < 1/\lambda < 0.1s \quad (5.11)$$

An average of 200 samples was taken for each of the 400 parameters distributed uniformly in the range given by (5.11). Figure 5.6, 5.7, and 5.8 are presented next to give an insight about the dependence of the autocorrelation function and the mean delay R.V with parameter $1/\lambda$. For small mean delay parameters

(e.g., $1/\lambda < 1e-4$), the autocorrelation function preserves its energy for the autocorrelation variable h in the range, $|h| < \text{Tone Length}$ (e.g., 8.2 s), see Figure 5.6.

If the mean delay parameter $1/\lambda$ is in the range, $0.1ms < 1/\lambda < 1ms$ the symmetric triangular autocorrelation of Figure 5.6 has a longer coherence time. The shapes for the correlation function in these ranges can be approximated by a lorentzian function as illustrated at Figure 5.7.

The autocorrelation dispersion has wider ranges for the correlation variable h as the mean delay parameter $1/\lambda$ grows above 1ms, for delays around 100 ms the shape of the autocorrelation function for a tone of 8 ms length and frequency of 555.3 Hz has the form presented if Figure 5.8.

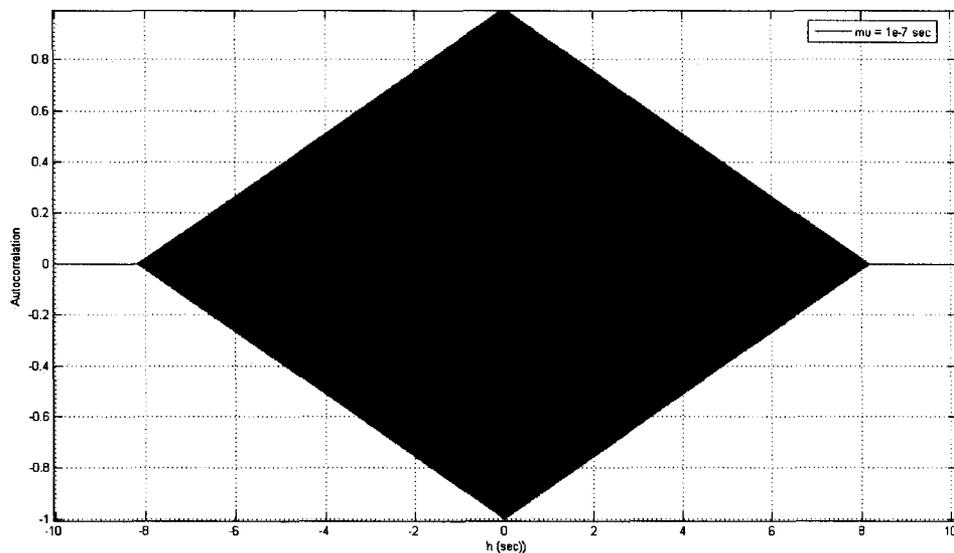


Figure 5.6: Autocorrelation function for an exponential R.V. with parameter $1/\lambda = 1E^{-7}$.

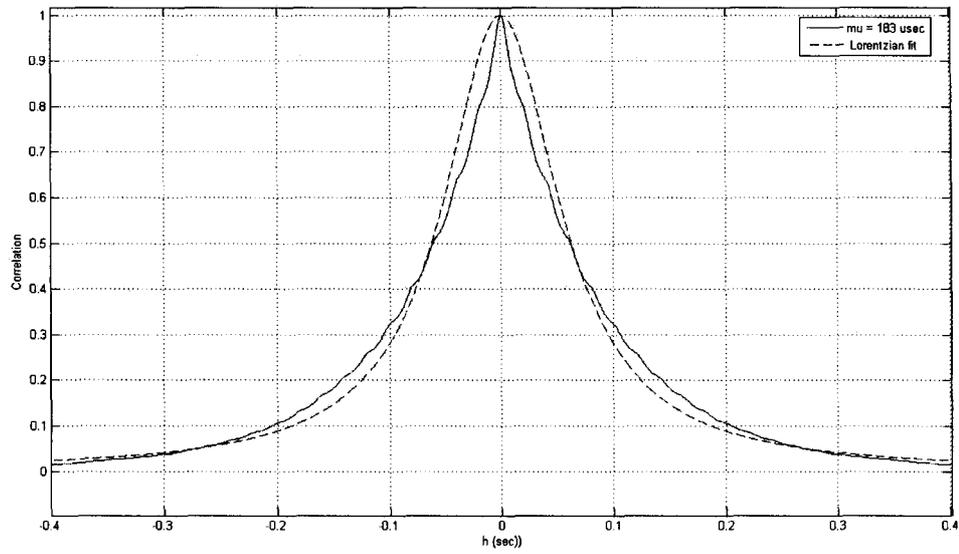


Figure 5.7: Autocorrelation function for an exponential R.V. with parameter $1/\lambda = 183\mu s$

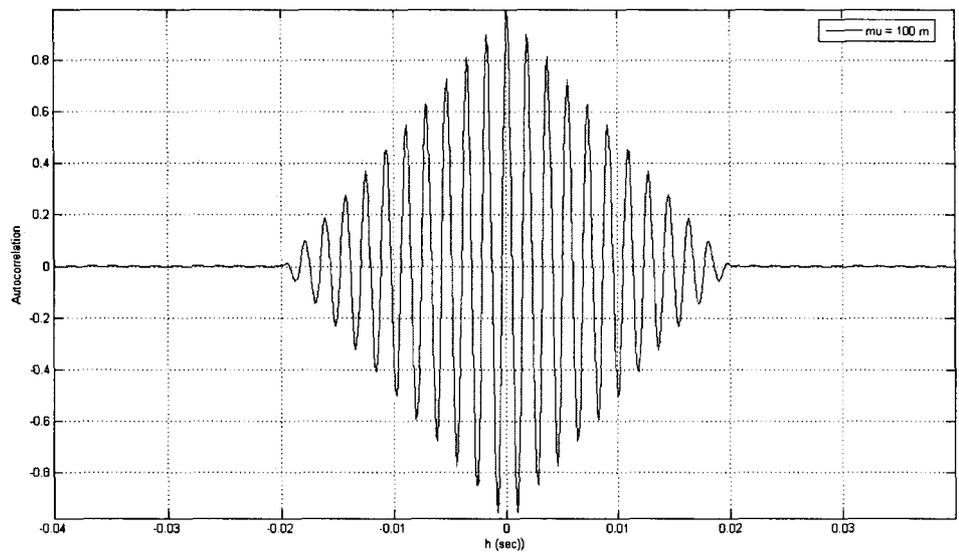


Figure 5.8: Autocorrelation function for an exponential R.V. with parameter $1/\lambda = 100ms$

Figure 5.9 shows the dispersion of the autocorrelation function along the autocorrelation parameter h . The simulation considers 7 different tone lengths (e.g., 32 ms, 64 ms, 128 ms, 256 ms, 512 ms, 1.024 s, and 2.048 s) for an exponential R.V with parameter $1/\lambda$ in the range (5.11).

Figure 5.10 describes how the correlation parameter's range h increases as the exponential Parameter $1/\lambda$ does; range k keeps its original length for $1/\lambda < 0.1ms$, the h increases linearly with the highest slope for the longest transmitted tone, (e.g., 2048 ms). As a single fragment of media duration of 20 ms does not depend of any R.V it remains its original range.

The Tone duration has effect in frequency domain over the power peak of the sampling pulses resulting from the Fourier Transforms of the triangular-lorentzian-triangular correlation functions. With the increase in tone duration the square sinc function concentrates its energy at smaller wide-line and the largest power peak is for the longest transmitted tone. See Figure 5.11.

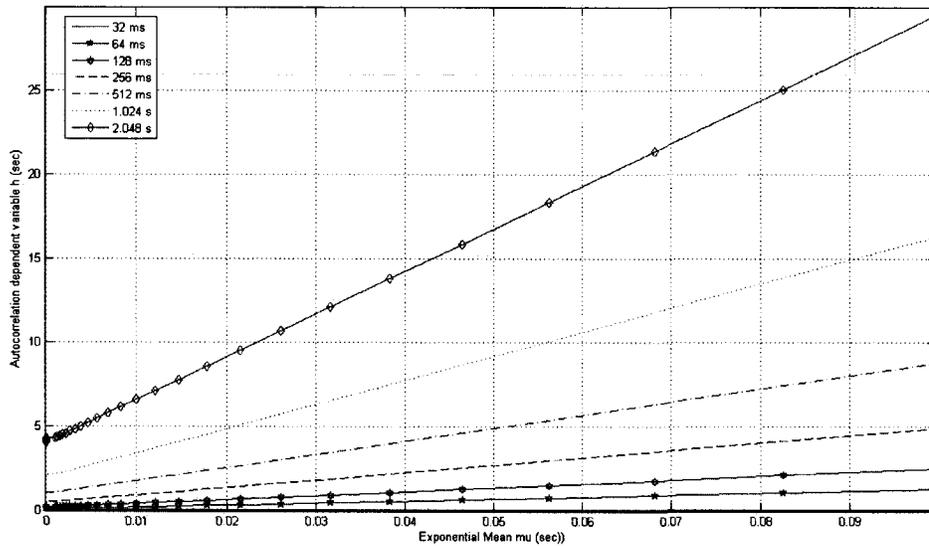


Figure 5.9: Autocorrelation parameter h dependence on the exponential R.V delay mean.

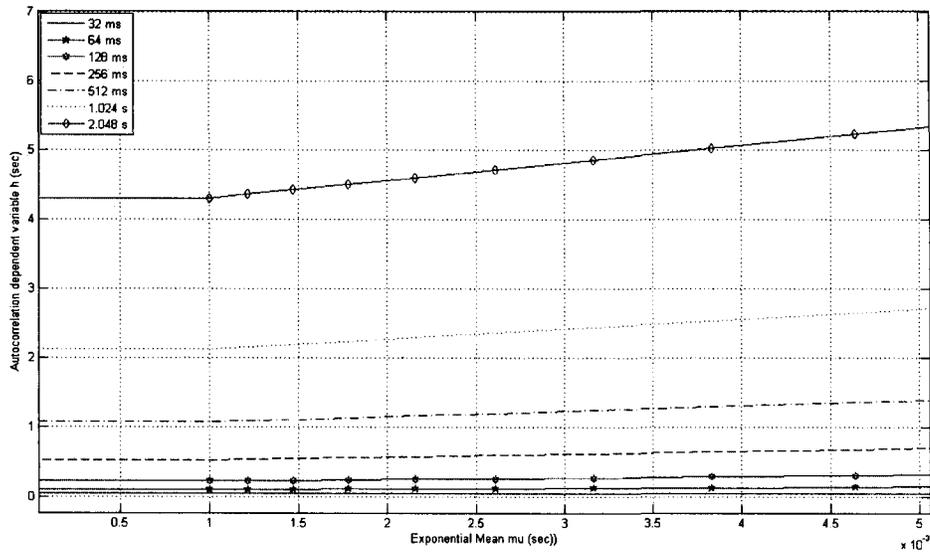


Figure 5.10: Autocorrelation parameter h dependence on the exponential R.V mean delay, this zoom shows the independence of the autocorrelation and the R.V. for very small means

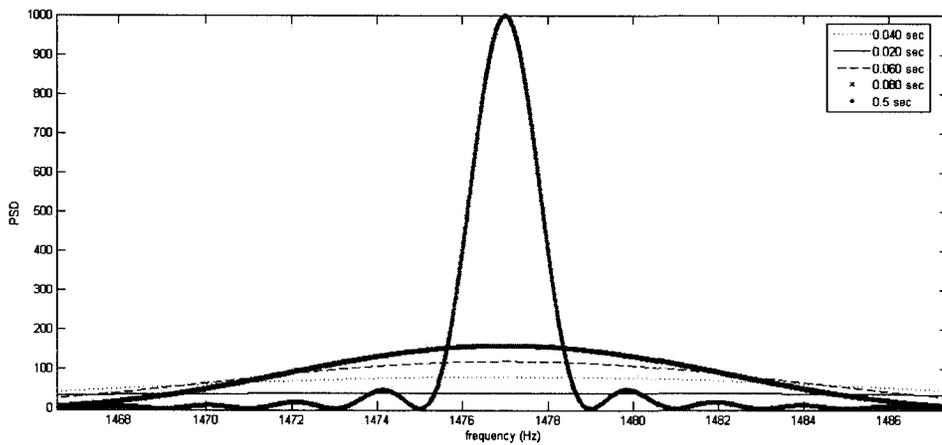


Figure 5.11: Power spectral density comparison, exponential distributed delay with parameter $1/\lambda < 0.1 \mu s$ and 5 tones duration. The tones lengths are specified at legend in figure. Here there is tones centered at 1477 Hz

The autocorrelation function for the transmission of two simultaneous tones at different frequencies is the superposition of each tone autocorrelation function. Which means it is possible to simulate 3 or 4 simultaneously transmitted frequencies.

5.5 Power Penalties Results

Increasing the tone duration of the tones to be transmitted is reflected in the power penalties with lower losses for the largest transmitted tone. Figure 5.12 is for tone duration of 8 seconds and compared with the longest tone of 2 seconds in Figure 5.13, there is 12 dB more losses for the shortest tone.

From figures 5.11 and 5.10 it is clear why the VoIP does not work without the inclusion of a de-jitter buffer even when no packet losses and out of sequences were considered.

5.5.1 Gaussian distributed R.V power penalties

The Power Penalties for normal distributed random variable with parameters in the range (5.12) and (5.13) are shown from Figure 5.14 to 5.16.

$$0.1.\mu s < \mu < 1ms \quad (5.12)$$

$$1\mu s < \sigma < 1ms \quad (5.13)$$

Mean and standard deviation are the same for the exponential distributed R.V, in the case of the Gaussian R.V parameters can be manipulated independently. In the case for the exponential R.V the highest penalties incurred for a mean delay variance $1/\lambda > 1ms$.

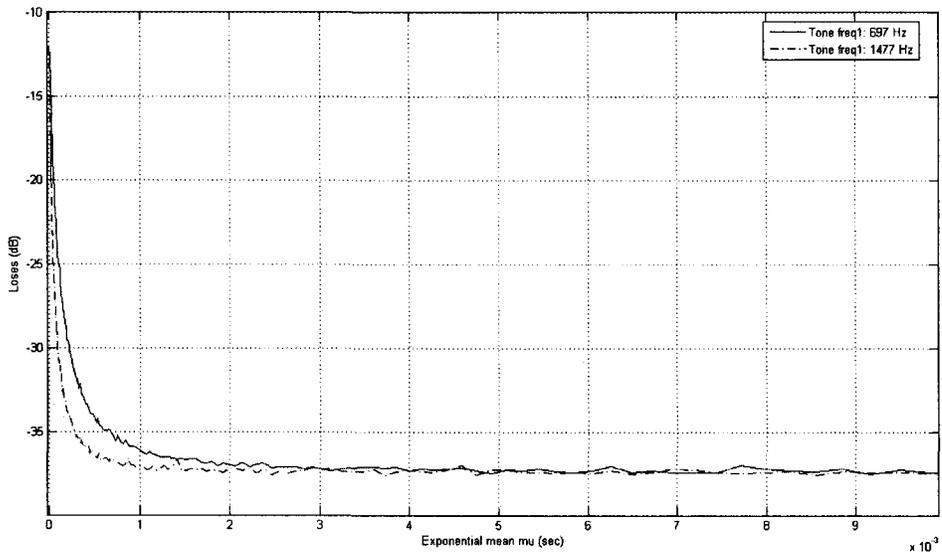


Figure 5.12: Power penalties for a DMTF signal with an exponential R.V mean delay in the range (5.11). No de-jitter buffer was considered in the simulation.

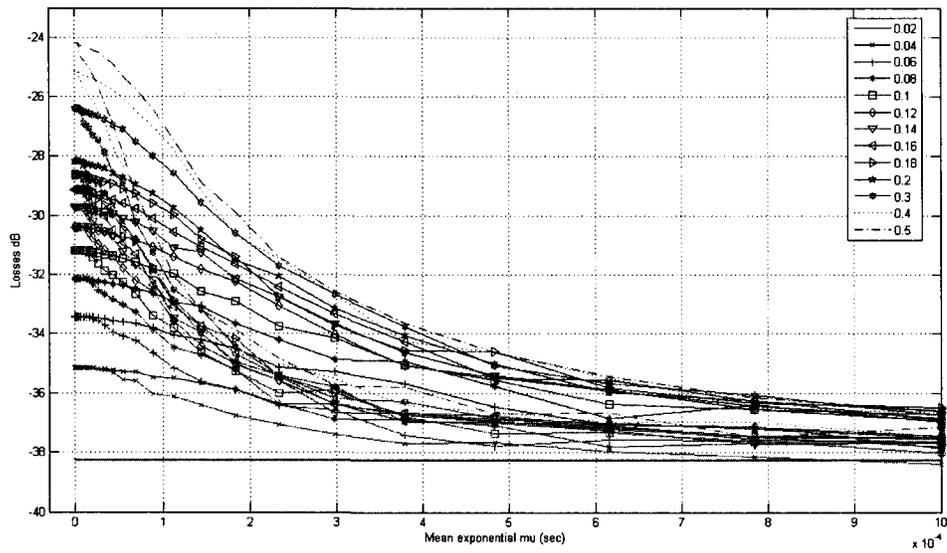


Figure 5.13: DMTF signal at frequencies 697 and 1477 Hz. Exponential mean 13 tones lengths specified at the legend in figure. The largest tone duration in time, the lowest power penalties it has.

Figure 5.15 shows the results of simulating normal distributed R.V with parameters given by (5.12) and (5.13). The legend at the figure indicates the value of the standard deviation σ and the correspondent line in the plot. From the results it can be observed the transition region of losses is for a range:

$$0.1ms < \sigma < 1ms \tag{5.14}$$

Then the relation of the R.V used for the simulation and the Power penalties is independent of the mean delay parameter and is only affected by the delay variance. It is well known that the most negative effect for VoIP speech quality is the jitter or delay variation, see Figure 5.16.

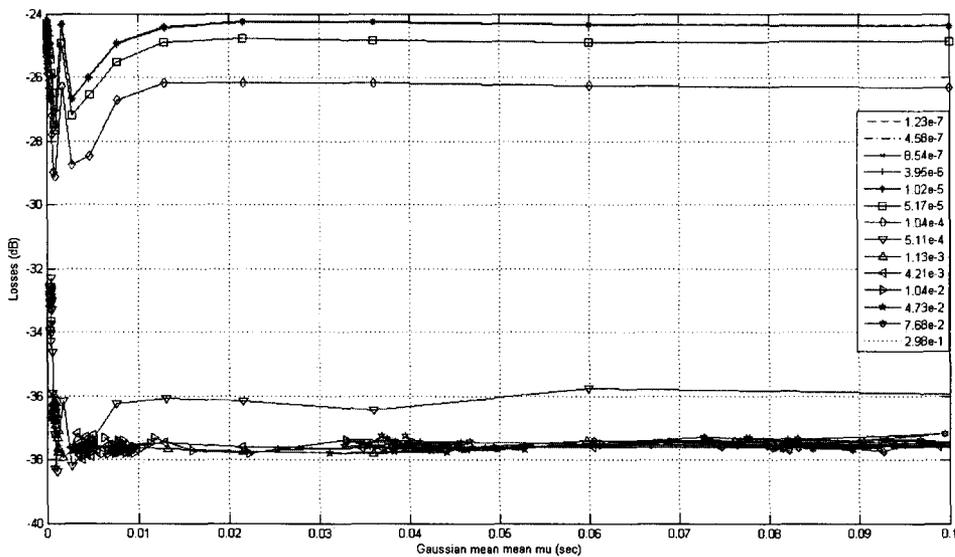


Figure 5.14: Power penalties comparison for a tone of 500ms length, mean delay normal distributed, and 14 standard deviations ($1e-7$ $5e-7$ $1e-6$ $5e-6$ $1e-5$ $5e-5$ $1e-4$ $5e-4$ $1e-3$ $5e-3$ $1e-2$ $5e-2$ $1e-1$ $5e-1$). The highest losses are for standard deviations larger than 1 ms.

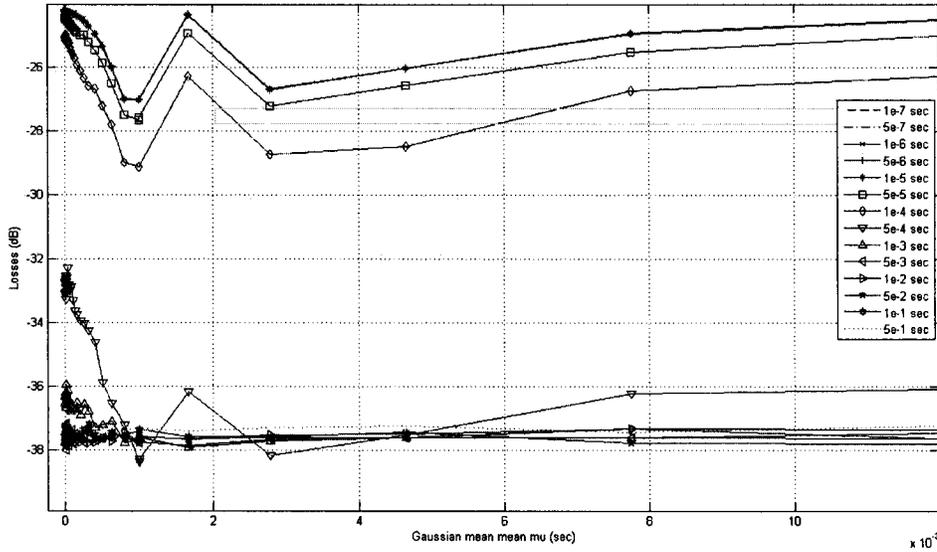


Figure 5.15: Power penalties comparison for a tone of 500ms length, mean delay normal distributed, and 14 standard deviations. An oscillation of less than 2 dB in Power penalties is appreciable for mean delay range of $1ms < \sigma < 3ms$, but less determinant compared with the losses incurred by delay variation.

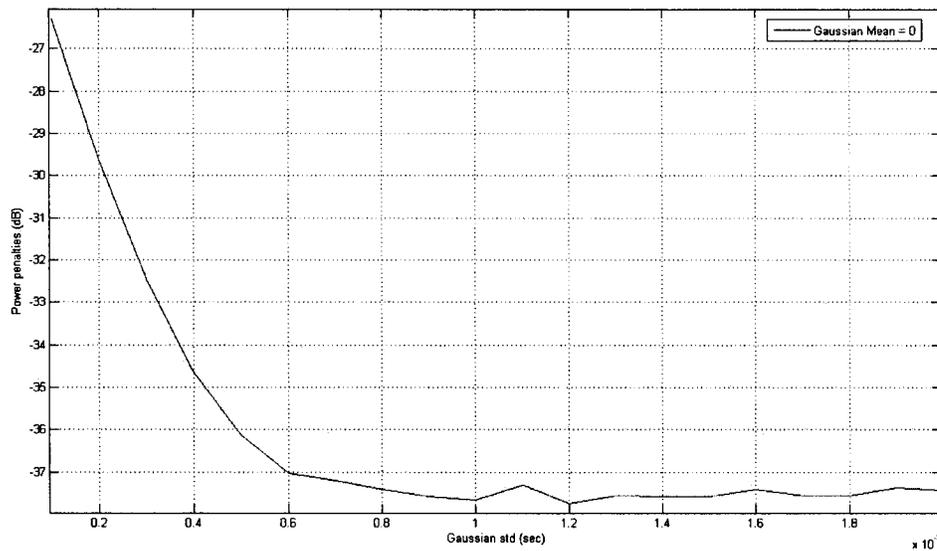


Figure 5.16: Power Penalties transition with the increment of parameter σ in the range (5.14).

5.6 Power Penalties analysis with a de-jitter buffer inclusion

Section 5.3 gives all the details about the process used to simulate a packet traveling from SLAN to DLAN by independent paths, and the way in which each packetized fragment of a tone is delayed and therefore possibly arrives to destination out of sequence in relation with the original media stream.

The results presented next show how the inclusion of a de-jitter buffer in a VoIP system improves significantly its performance. The cost is some lost packets, but last chapter showed how lost packets have lower impact on the reconstructed media stream than the variation of the delay from SLAN to DLAN.

Next are presented the results from simulating an exponential R.V using 5.1 and 5.9 for a range of the R.V parameter:

$$0.1\mu s < 1/\lambda < 500ms \quad (5.15)$$

Figure 5.17 shows a comparison in power penalties for the same exponential R.V, but the one with lower power penalties has been the result of processing that amount of fragmented and packetized audio stream by a de-jitter buffer with parameters shown at the legend in Figure 5.17.

De-jitter buffer is clearly one of the important components that makes VoIP system a functional technology. Power penalties for the same signal are analyzed under criterion (5.1) and (5.2) in comparison with the one build with (5.1) and (5.9) processed by a de-jitter buffer with SQT=400 ms and a Buffer size of 400 ms, shows an improvement in power of more than 20 dB.

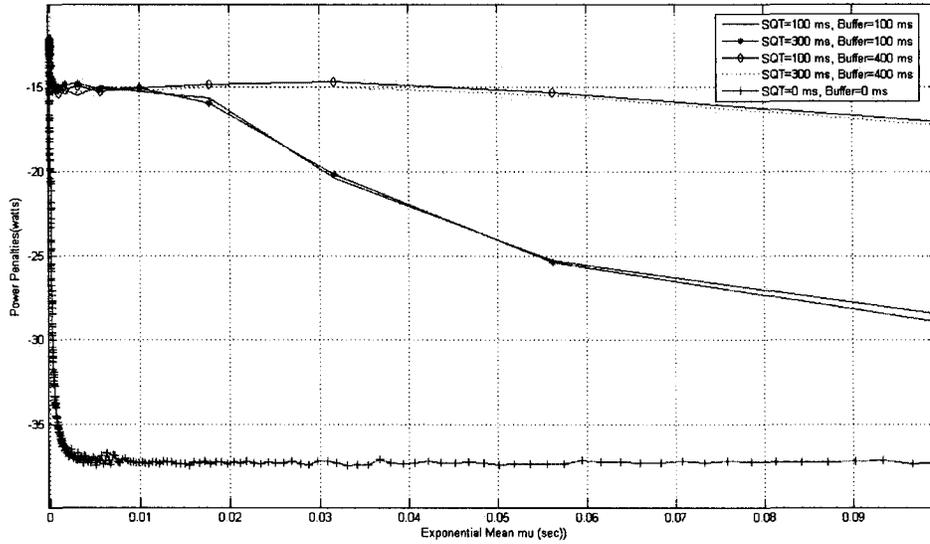


Figure 5.17: Power Penalties comparison for an exponential distributed R.V in the range (5.11). The dotted line with specifications at legend has the best Power Penalties.

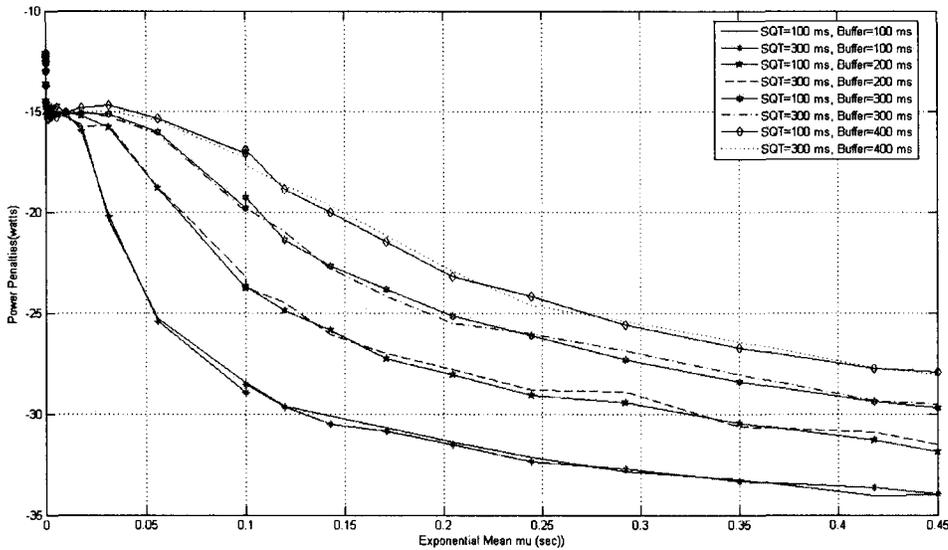


Figure 5.18: Power Penalties comparison for an exponential distributed R.V distributed in the range (5.15). The SQT factor does not have any considerable effect on Power Penalties.

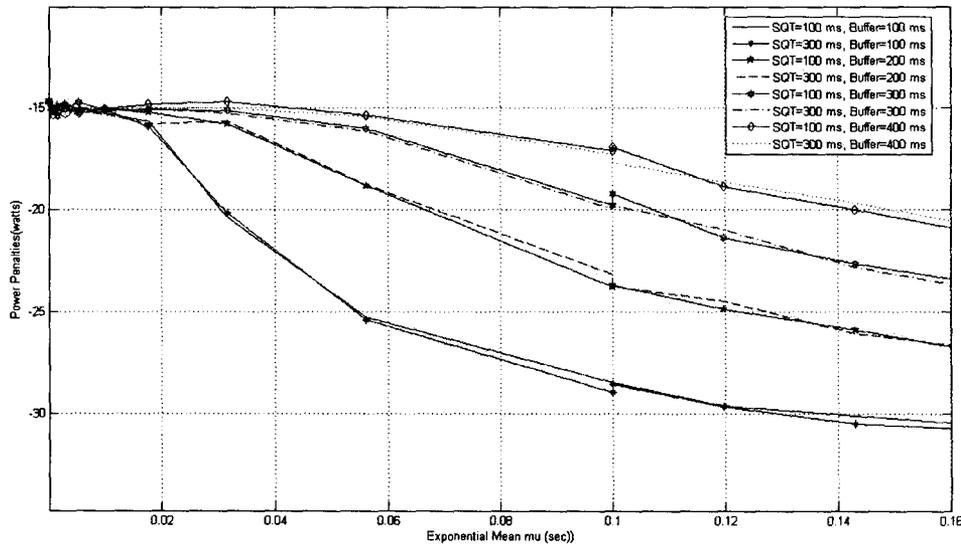


Figure 5.19: Power Penalties from a stream of packetized fragments exponentially delayed transmitted tone with mean delay R.V ranging in (5.14).

For large mean delays, the system improves its Power Penalties in 4:2 dB for each extra buffer time added of 100 ms. From Figure 5.19 for a Buffer size of 400 ms and a mean delay of 120 ms it has been penalized 3 dB of power. Augmenting 100 ms the size of the Buffer could be enough to increase the mean delay tolerance to 150 ms.

For large means the best improvement in the system performance (e.g., between Buffer sizes 100-400 ms) is 7.5 dB approximately, then improves evenly until 100 ms (e.g., 10 dB).

The buffer size does not have any impact on system performance for extremely small means, ($1/\lambda < 20ms$).

Figures 5.15-5.17 are also meaningful for modeling the standard deviation effect for exponential R.V, because the mean and the variance are the same variable.

5.7 Linewidth analysis

In the last section the Power Penalties suffered by a packetized fragments for a frequency tone are described as well as the positive effects of a de-jitter buffer with two degrees of freedom, the SQT and the Buffer size. The effect on the linewidth of the transmitted pulses is presented next.

First the line-width is presented for a packetized exponential delayed tone of 8.2 seconds long in the range (5.11) and no buffer implemented. From Figure 5.8 it was observed how the autocorrelation around zero disperses along the autocorrelation parameter h for large mean delays.

The rapid limitation in time for the autocorrelation function causes on the PSD pulses a Power Penalty, because the energy dispersion in pulses of the PSD. Our simulations consider fragments of 20 ms of packetized audio stream, which expands rapidly its correlation function along the correlation variable h ; this is the cause of the rapid losses of Power of the last section.

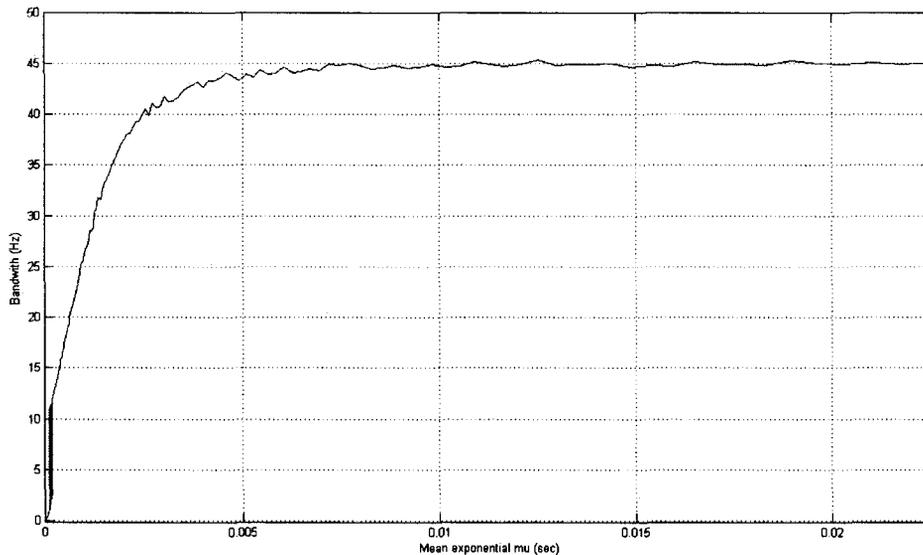


Figure 5.20: Bandwidth against exponential delayed plot for a tone of 8.2 s length implemented without buffer. The plot shows how the bandwidth of the pulse grows for exponential means below 10ms, then the pulse bandwidth remains constant in 45 Hz for exponential means above this value.

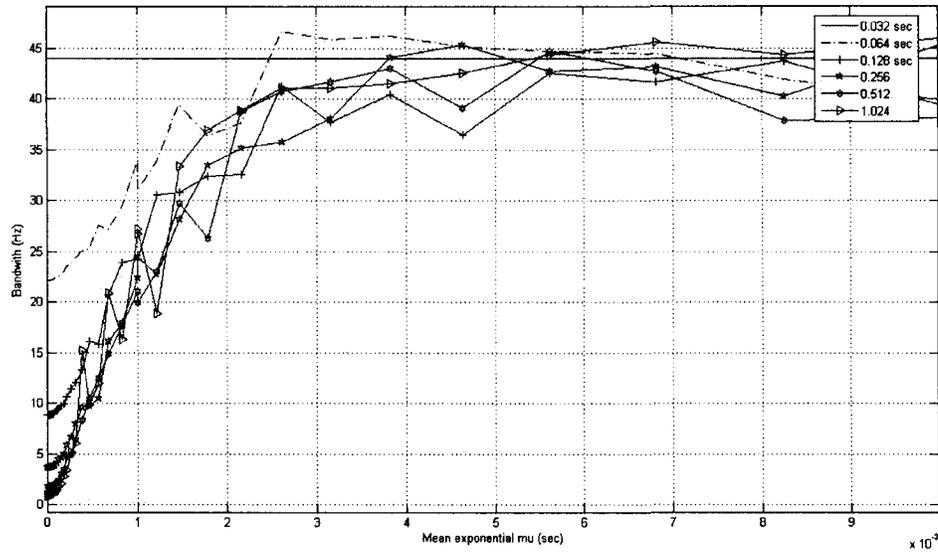


Figure 5.21: Line-width tone variation for different tone lengths Fragmented and Packetized. Exponential R.V with parameter $1/\lambda$ in (5.11). It can be seen the rapid dispersion of the pulses along the spectrum. A system like this would be useful only for mean delay parameter $1/\lambda < 0.1ms$.

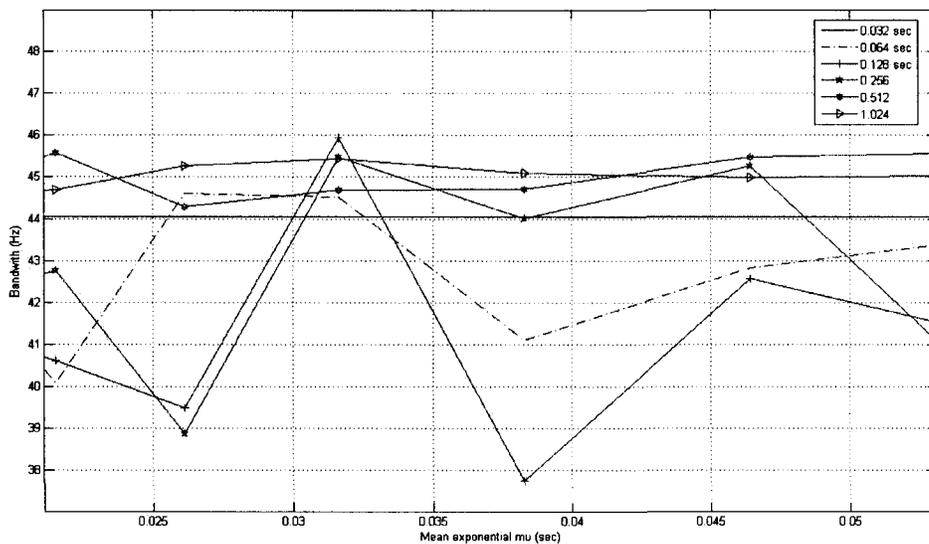


Figure 5.22: Tone linewidth variation for medium tone lengths exponentially delayed (e.g., $0.02s < 1/\lambda < 0.5 s$), tones are approximately stable for very short durations (e.g., 20 ms), and for tone durations above 500 ms.

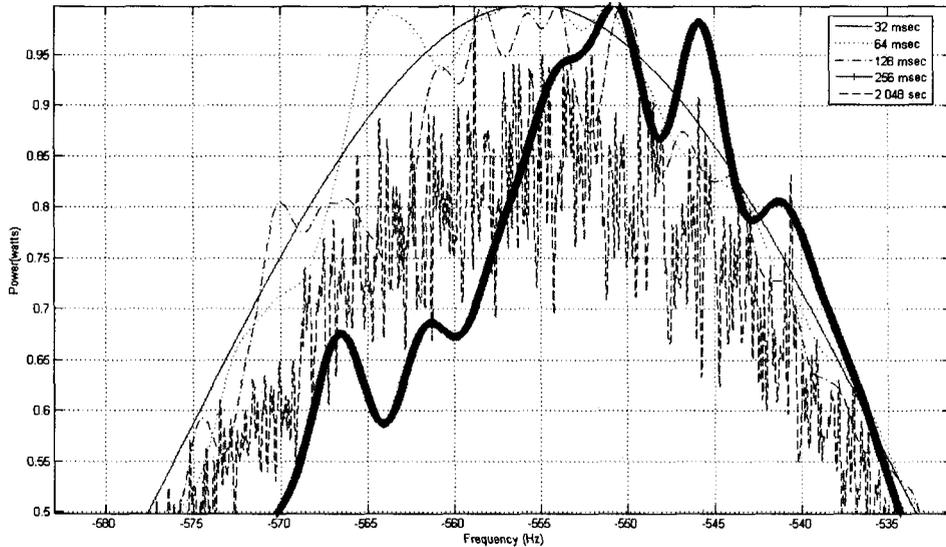


Figure 5.23: The cleanest and noisiest power spectral density signals plotted above keeps the original transmitted pulse bandwidth. The pulses correspond to tone lengths in time of 32 ms and 2.048 s, respectively.

Figure 5.23 shows how shorter duration time pulses suffer more distortion, which originates noise in linewidth measurements.

5.8 Linewidth analysis with a de-jitter buffer inclusion.

From the last section analysis we see the expansion in frequency domain of the square sinc pulses of the PSD. This effect causes Power Penalties restricting the last modeled system to have an acceptable performance in scenarios with an exponential R.V. below $1/\lambda < 0.1ms$. Actual VoIP needs tolerances around 150 ms, which can be reached as shown by Figure 5.19 with the inclusion of the right size Buffer and SQT.

Figure 5.24 shows the results of simulating an exponential delayed Tone of 8.2 s seconds length processed by a de-jitter buffer of parameters in the range:

$$100ms < Buffer_Size < 400ms \tag{5.16}$$

$$100ms < SQT < 500ms \tag{5.17}$$

The results from Figure 5.24 show a very considerable reduction in the line-width needed for the pulses from the Power Spectral Density of (5.1), from 45 Hz to a range of (3-8 Hz) for large mean delays.

It can be observed from Figure 5.24 the effect of the buffer on the line-width of the pulses. From Buffer sizes 100 ms - 200 ms the bandwidth for pulses in a 100 ms buffer is double compared with the pulses de-jittered in a 200 ms buffer. The linewidth variation for pulses de-jittered in buffers of 200 ms-300 ms is lower (e.g., 1.5 Hz). Therefore, Buffer reduces the improvement in the VoIP system performance as the Buffer Size increase in size.

De-jitter buffer helps to reduce the maximum delay variation or jitter for the system, it translates to a physical effect on the signal related to the linewidth of the frequency pulses packetized and transmitted over IP/UDP/RTP (e.g., see Figure 5.21).

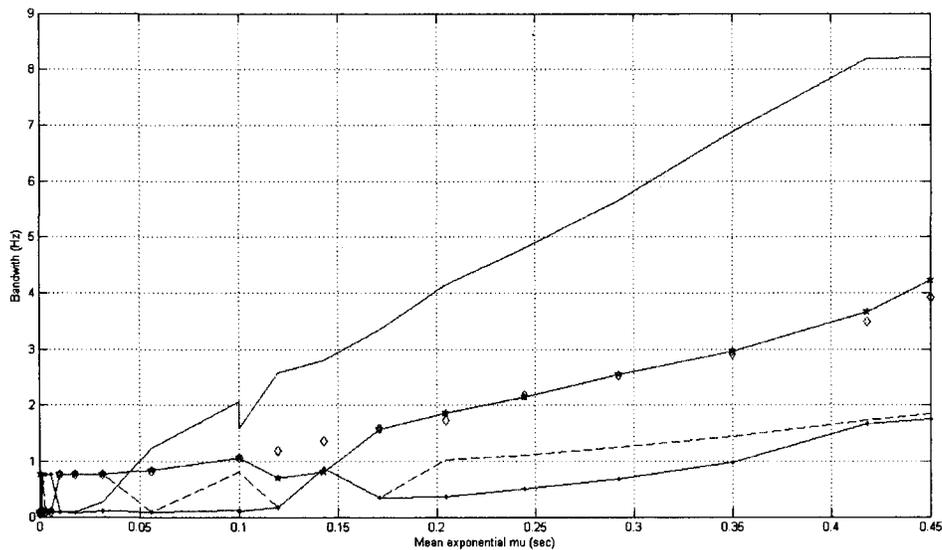


Figure 5.24: Bandwidth for an exponential delayed Tone of 8 s, the effects on the

line-width of 5.1 are simulated for a de-jitter with parameters in the Figure legend..

5.9 Advantage of including at the de-jitter buffer in a VoIP communication

We shall now see how the Buffer changed the mean and variance of our packetized media stream, as observed in Figure 5.16 there is a range for delay variance where the Power Penalties are so high. Figure 5.25 is a plot for the packet playtime delay against the original delay for an exponential R.V. The graph has an offset of 20 ms, which is the natural delay for the source to pronounce the fragment of media to be packetized.

5.10 Mean Delay Playtime and Mean Jitter Playtime

From Figure 5.26 it is identified the 150 ms zone to have a new packet playtime delay of 1 ms, a significant amount to make VoIP a functional technology, the most destructive noise is the MDJ, see Figure 5.28 for the 100-150 ms zone.

From Figure 5.28 the MDJ in the zone from 100-150 ms is in the region of transition to highest Power Penalties region, The mean delay packet playtime distribution has less sensitivity with mean jitter playtime, however is possible to add 100 ms of dejitter buffer and analyze if higher delays are supported by the system.

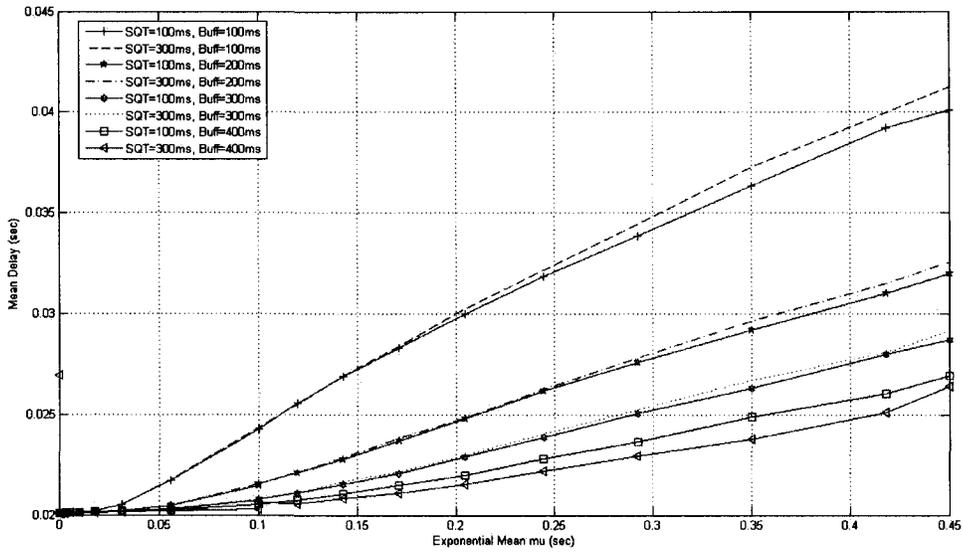


Figure 5.25: Mean packet Delay playtime after buffer

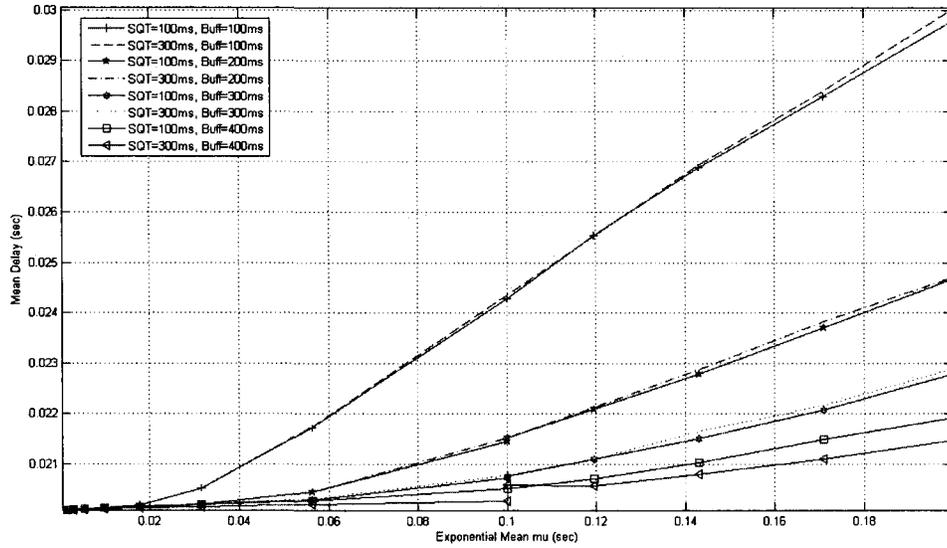


Figure 5.26: A better approximation of the Mean packet playtime to observe the 100-150 ms zone.

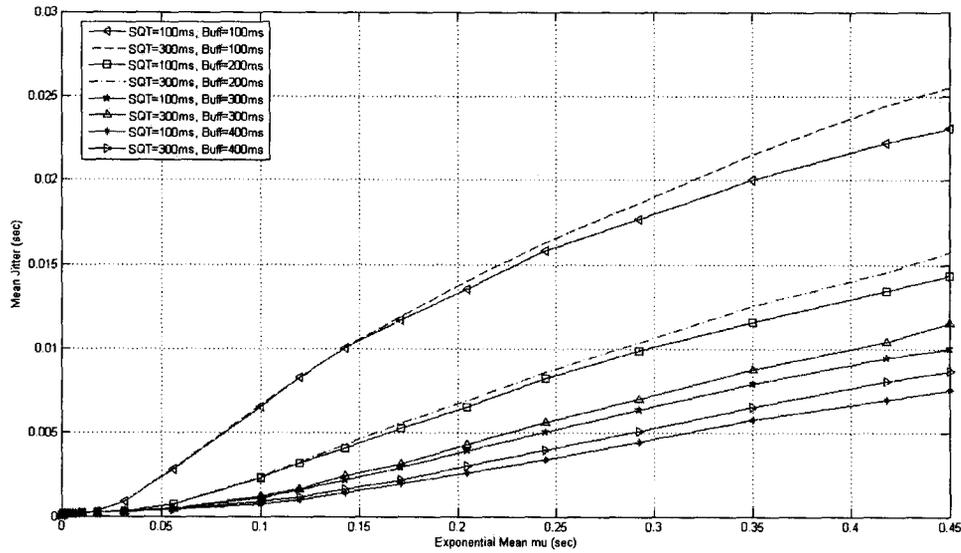


Figure 5.27: Mean packet playtime jitter for an originally exponential R.V with parameter (5.15).

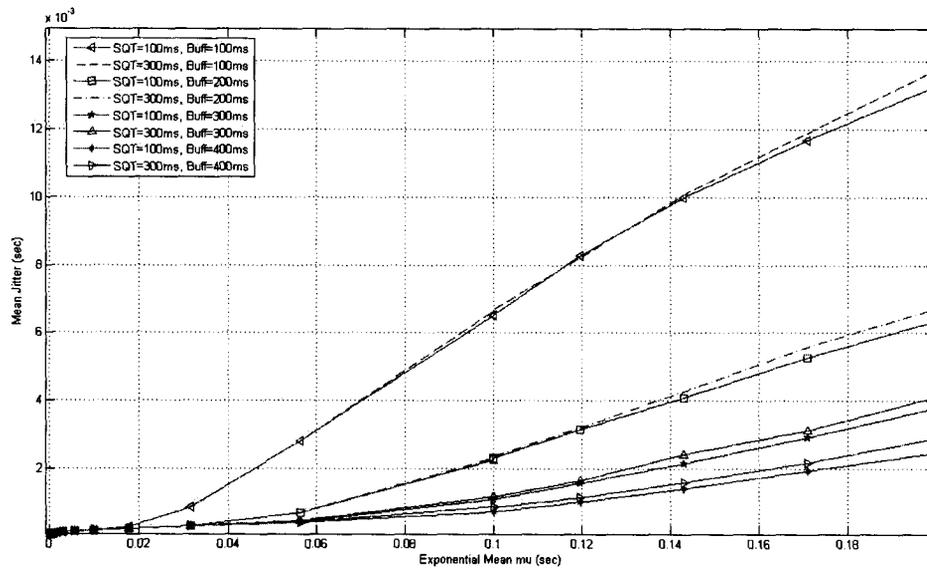


Figure 5.28: A better approximation of the Mean packet playtime jitter to observe the 100-150 ms zone.

5.11 Mean Lost Packets and Delay Variation playtime

In [12], it was proposed to analyze the impact of packet losses, showing that the same packetized media stream could have different QoS measurements resulting for the same frame of packets but distributed in different ways into the 20 ms slots. Good techniques for sending packets in a way such that burst packets losses are avoided are commonly applied. Also it could be useful to sent redundant packets for the most important ones.

The lost of packets resultant from the system described by (5.1) and (5.9) with mean delay exponential R.V in the range (5.15). Buffers of 300-400 ms and exponential means delays of ~ 150 ms the packet losses are less than 5% of a total of 405 packets. Then we can say for a transmission with an exponential delay with mean below 120 ms has a mean packet losses of less than 1.2 % approximately, see Figure 5.30.

The first order estimator for the calculation of mean delay jitter used was the one provided by [15] originally extracted from [18] is now compared with the main delay variation playtime in the form of 2.12. It can be compared from Figure 5.28 and Figure 5.32 that the estimator proposed in [15] to calculate jitter is more flexible than mean delay variation from Figure 5.32. In the region (5.19) the mean delay jitter packet playtime is below 1 ms while the mean delay variation is around 8 ms.

From Figure 5.32 can be appreciate the impact of the Buffer parameter SQT on the mean delay variation of the VoIP modeled system. For mean delay variations around 140 ms the system has a range for mean delay variation packet playtime of 7-10 ms for buffer parameters SQT in the range 100-500 ms. Simulations have shown that an SQT factor of 0 results in a failed VoIP communication.

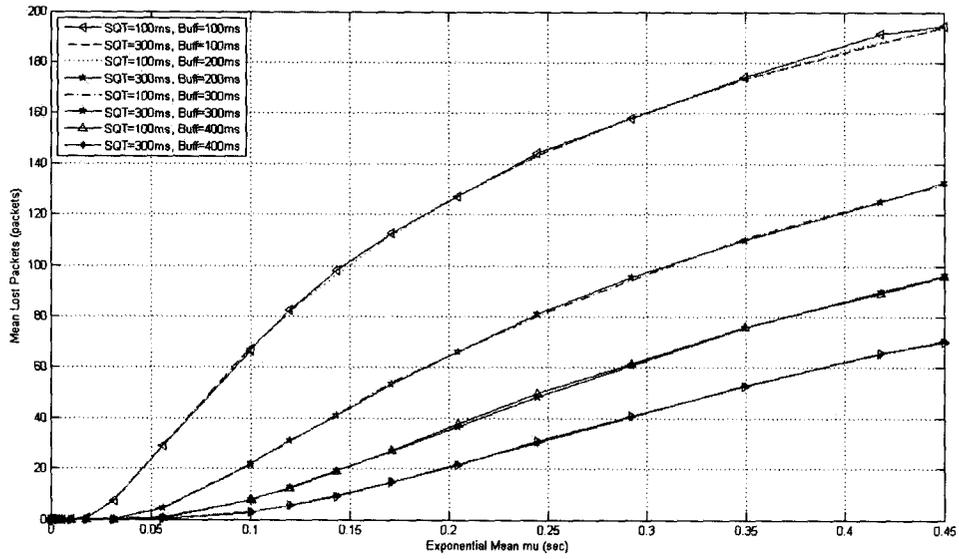


Figure 5.29: Mean Packet Losses for an exponential distributed R.V with mean delay in the range (5.15)

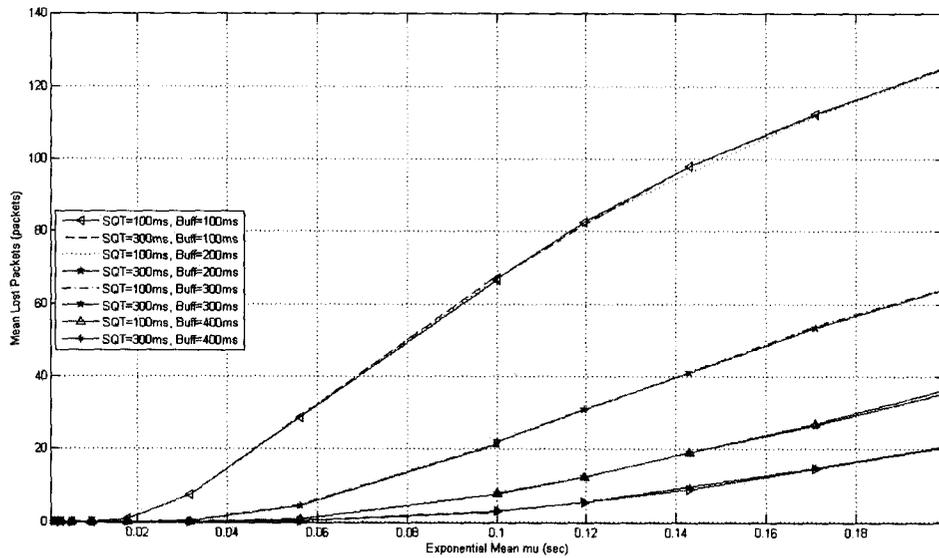


Figure 5.30: Mean Packet Losses for an exponential distributed R.V with mean delay.

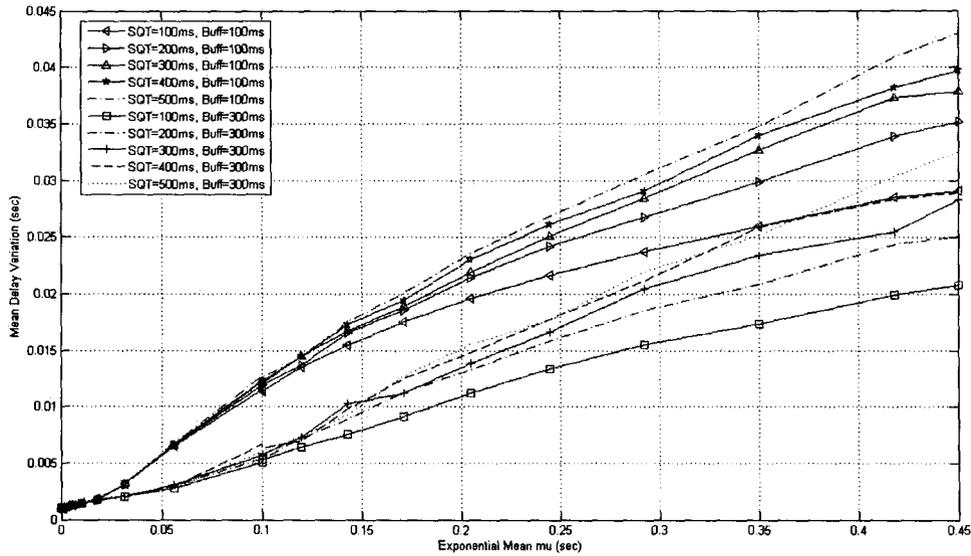


Figure 5.31 Mean Delay variation for an exponential distributed R.V with mean delay in the range (5.15)

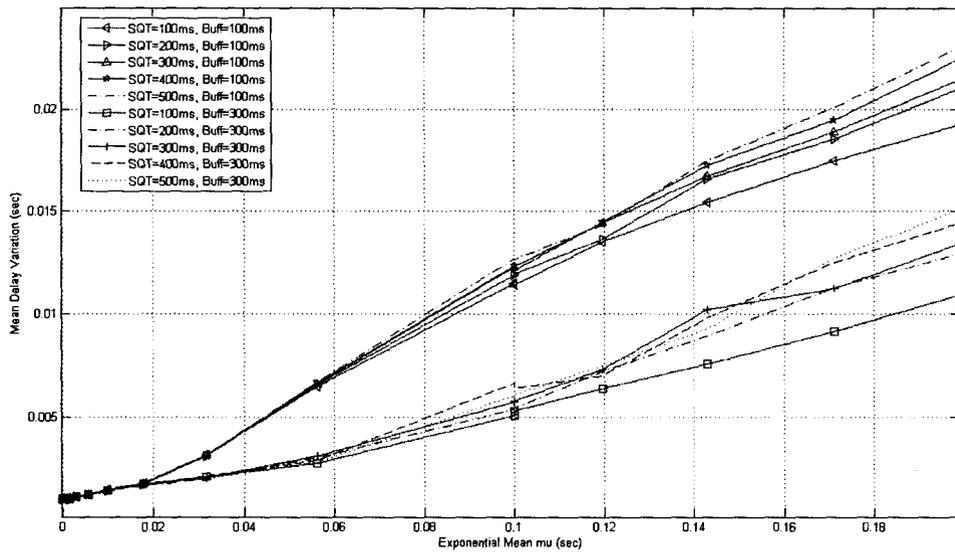


Figure 5.32 Mean Delay variation for an exponential distributed R.V with mean delay in the range (5.15)

5.12 Packet Loss and Mean delay playtime Jitter relation with Buffer parameters (SQT and Buffer size)

Figure 5.33 compares the mean packet losses with respect to the buffer size, and from Figure 5.34 a comparison among the packet losses dependence with respect to SQT is presented. Both graphs were simulated using a Buffer Size of 400 ms and 600 ms. From the results of these graphs it can be concluded that bigger buffers in size help to reduce the mean number of lost packets. The simulations were done for an exponential R.V with mean delay parameter of 100, 200 and 300 ms. because the delay variance for exponential distributions is the same as the mean delay; the curves show an increment in lost packets for larger mean delays, but is not the effect of the mean delay the cause of losses, the mean packet losses are a penalty rising from delay variance, whereas packet losses seem to be independent of SQT.

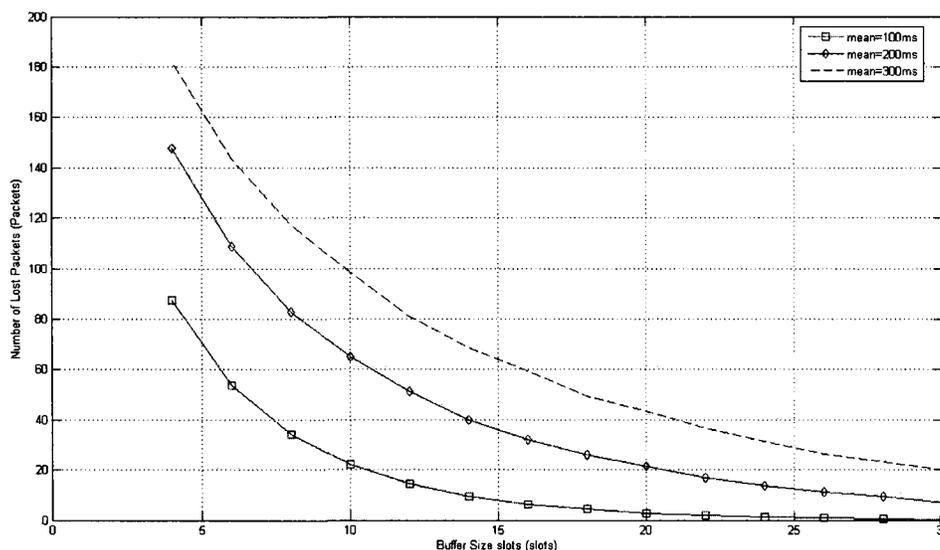


Figure 5.33: Number of Packet Losses dependence with Buffer size.

Figure 5.35 shows the mean packet delay for buffers sizes of 30 and 40 slots of time; each slot is 20 ms length, and mean delays for network of 200 and 300 ms, specified at the legend of the Figure. The simulation results show how larger network delays (e.g., 300 ms) cause the mean delay playtime jitter to grow for a larger SQT range, this can be seen from plots for a network delay of 200 ms, and where SQT values larger than 300 ms does not cause more changes in the mean delay playtime jitter.

In Figure 5.35, it is noticed that in the zone for $20\text{ms} < \text{SQT} < 200\text{ ms}$ for mean delays of 200 ms the mean delay jitter grows fast until it reaches it maximum and becomes independent of SQT. Note how 300 ms of mean delay in network provoke the mean delay jitter playtime to stabilize for $\text{SQT} > 600\text{ ms}$.

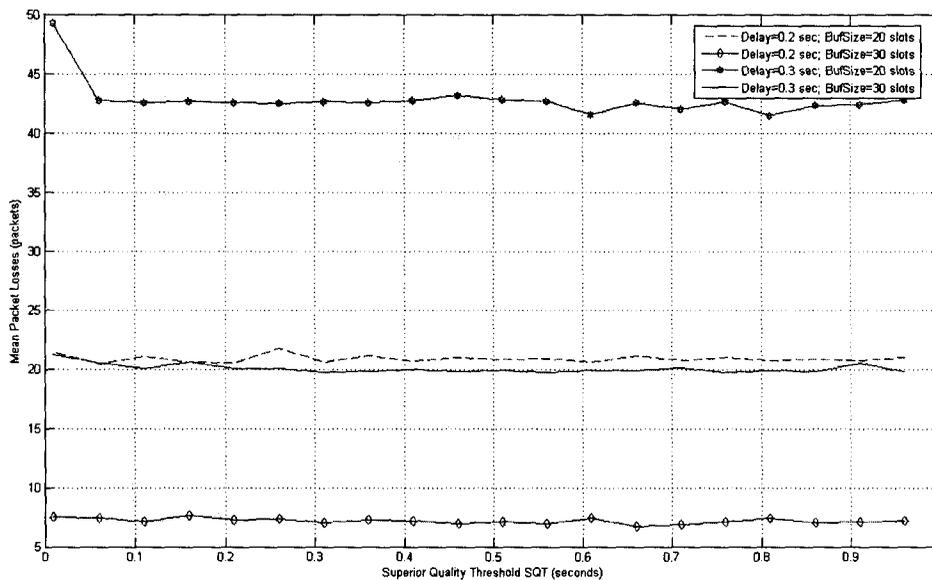


Figure 5.34: Mean Packet Losses SQT independence for $\text{SQT} > 20\text{ms}$.

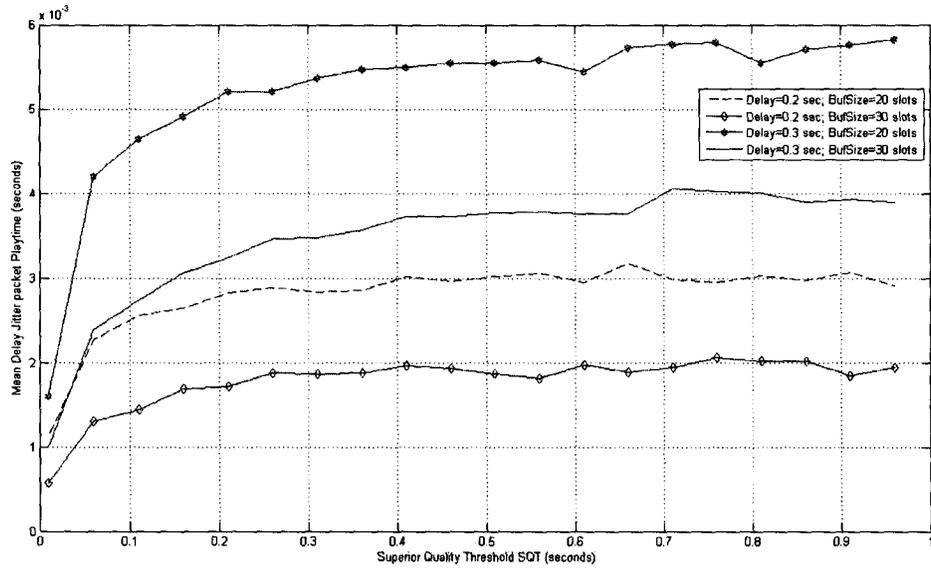


Figure 5.35: Mean Delay Jitter Playtime dependence with SQT.

CHAPTER VI CONCLUSION AND FUTURE JOBS

First of all, it was shown how the perceived QoS in some cases will not depend only on the objective parameters describing the state of a given network, but also on the type of services available in the network. In current VoIP networks and in future Next-Generation Networks, address convergence is a must. The viability for a user to perform an ENUM query through a client operating the ENUM protocol is the standardized approach for address convergence and hence, a very important factor directly affecting the perceived quality of the network. Clearly, if a user can initiate a VoIP call or any other communication session by just knowing the e164 number of the called party, his perception of service quality is enhanced. The importance of implementing an ENUM provisioning system on every considerable region in the world has been highlighted; this will allow developing all the capabilities provided by VoIP in terms of the applications and services possible through this technology. An ENUM trial, in conjunction with NIC Mexico, was developed starting from scratch, and even, an ENUM client was developed for a cellular CDMA company.

Secondly, some advantages of sampling a VoIP communication in real time have been analyzed. The viability of describing the expected QoS of VoIP call by extracting statistical measurements about the state of the Network was proved by implementing a QoS monitor that basically calls a PBX, records a pre-recorded message for PESQ comparison and in parallel captures the RTP data. Some advantages of sampling in real time a packetized media stream are the possibility of tracking the evolution of traffic in a given network, and how this affects the quality of the transmitted voice stream.

Other important advantage of sampling in real time a VoIP media stream is the possibility of analyzing real scenarios, and using the method proposed in chapter 5, to generate an estimation of the perceived speech quality. The media stream can be simulated through different buffer configurations, therefore it is possible to determine the cause originating the penalties in the VoIP communication, (e.g., an excessive delay operating on the network or a wrong dimensioned de-jitter buffer).

The effectiveness of a novel method used to describe the impairments suffered by a VoIP communication has been shown. The model is able to map the objective parameters degrading a given network in the form of mean packet delay and mean delay jitter to Power Penalties, line-width spreading, and mean delay playtime jitter.

Results showed that the Power Penalties are independent from the mean delay for both, exponential and Normal R.V's.

For many cases simulated, results showed that for standard deviations in the range $1ms < \sigma$, power is highly penalized. The conclusion is that a VoIP network should be designed to operate with a jitter less than 0.1 ms to avoid loss of signal strength.

The result of including a de-jitter buffer at destination in the analysis of a VoIP communication system, is the network jitter transition to an effective jitter after playtime to a lower value, desirable to be below 0.1 ms, see Figure 5.28.

The effect of buffer size on the mean number of lost packets was observed, it decreases as the buffer size increases. Nevertheless increasing buffer size indiscriminately is worthless because the improvements in the number of lost packets, which are acceptable in the range 2-5 percent depending on the used codec, are insignificant with respect of the Buffer cost.

The difference between the normal and exponential R.V with the same mean delay parameter is fixed by its standard deviation, for exponential and Gaussian case the resulting effective mean delay must be $1/\lambda < 1ms$ for the VoIP system to have an acceptable performance. When no buffer is included the performance of

the system is determined by the standard deviation in the Normal case and by the mean in exponential case, which must be in the same range $1/\lambda < 1ms$.

From Figure 5.34 we can observe the independence between Packet Losses and the buffer parameter SQT, where no change in packet losses is observed for any SQT value. The SQT has a stronger impact on mean jitter packet playtime if values are $SQT < 200$ ms, above this value there is smaller changes for the mean delay jitter.

In section 5.2 a Poisson Process was included to note the similarities with the process constructed by (5.1). The points marking the inter-packet playtimes could be written in the form of (5.6) with known autocorrelation. Note the process formed by (5.1) has its own restrictions of falling time after T ms, (e.g., 20 ms). Another insight is the shape in time the autocorrelation takes along parameter h , see Figures 5.6-5.8. Also is needed to find a theorem to relate the signal resulting of processing the autocorrelation of signal in (5.6) with the autocorrelation of the signal generated by the filter in (5.6). The objective is to use these assumptions to find an approximation for the autocorrelation in (5.1) which will describe the way IP packets are delayed and penalized depending on the state of the network.

The method proposed in Chapter V allows a description of the effects degrading the quality for a packetized audio stream due to the state of the network. This method allows the study of the impairments degrading the signal by its power spectral density which clearly help to understand the degradation incurred on the signal by pulse dispersion. The method also has the characteristics of been able to model the penalties affecting a packetized media processed by different buffer configurations to compare the results obtained by each one. The results for exponential and normal distributed random variables were obtained by the same procedure and its results were compared to provide some important facts to consider when designing a VoIP communications system.

REFERENCES

- [1] Akira Takahashi and Hideaki Yoshino, Perceptual QoS Assessment Technologies for VoIP, NTT Service Integration Laboratories Nobuhiko Kitawaki, University of Tsukuba, *ieee*, July 2004.
- [2]. M P Hollier, PA Barrett, Non-Intrusive perceptual quality measurement for quality assurance in NGN and 3G Networks, Psytechnics Limited, UK, *ieee*, June 2003.
- [3] Antony W. Rix, Perceptual speech quality assessment – A review, Psytechnics Limited, UK, *ieee*, May 2004.
- [4] Matti Karjalainen , A new auditory model for the evaluation of sound quality of audio systems, Helsinki University of Technology Acoustics Lab., Otakaari 5 A 02150 Espoo 15, Finland, 1985,
- [5] Objective quality measurement of telephone-band (300– 3400 Hz) speech codecs. ITU-T Rec. P.861, Feb. 1998.
- [6] Voran, S. Objective estimation of perceived speech quality – part I: development of the measuring normalizing block technique. *IEEE Trans. Speech and Audio Processing*, 7 (4), 371–382, July 1999.
- [7] Wang, S., Sekey, A. and Gersho, A., An objective measure for predicting subjective quality of speech coders., *IEEE Journal of Selected Areas in Communications*, 10 (5), 819– 829, 1992.

- [8] Rix, A. W. and Hollier, M. P., The perceptual analysis measurement system for robust end-to-end speech quality assessment., IEEE ICASSP, Vol. 3, 1515–1518, June 2000.
- [9] Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. ITU-T Rec. P.862, Feb. 2001.
- [10] A Passive Method for Monitoring Voice-over-IP Call Quality with ITU-T Objective Speech Quality Measurement Methods, ieee, 2002.
- [11] Hiroki Furuya, Shinichi Nomoto, Hideaki Yamada, Norihiro Fukumoto, and Fumiaki Sugaya, Experimental Investigation of the Relationship between IP Network Performances and Speech Quality of VoIP, KDDI R&D Laboratories, Inc. 2-1-15 Ohara, Kamihkuoka-shi, Saitama 356-8502, Japan E-mail: hirof@kddilabs.jp, ieee, 2003.
- [12] Florian Hammer, Peter Reichl, Thomas Ziegler, Where Packet Traces Meet Speech Samples: An Instrumental Approach to Perceptual QoS Evaluation of VoIP, Telecommunications Research Center (ftw.) Vienna Email: hammer@ftw.at, reichl@ftw.at, ziegler@ftw.at, ieee, 2004.
- [13] V. Raisanen, G. Grotefeld, Network performance measurement with periodic streams, RFC. 3432, IETF Network Working Group, November, 2002.
- [14] I G. Almes, S. Kalidindi P, A One-way Delay Metric for IPPM, RFC 2679, IETF Network Working Group, September 1999.
- [15] C. Demichelis, P. Chimento, Packet Delay Variation Metric for IPPM, IETF Network Working Group , RFC 3393, November 2002.
- [16] G. Almes, A One-way Packet Loss Metric for IPPM, IETF Network Working Group, RFC 2680, September 1999.

[17] V. Paxson, G. Almes, Framework for IP Performance Metrics, IETF Network Working Group, RFC 2330, May 1998.

[18] CADZOW, James A., Foundations of signal processing and data analysis, Macmillan Publishing Company, New York, 1987,

[19] RTP: A Transport Protocol for Real-Time Applications rfc: 3550

[20] ITU-T recommendation P.56 OBJECTIVE MEASUREMENT OF ACTIVE SPEECH LEVEL, ITU-T, Geneva, 1993,

[21] Papoullis, A., Probability, Random Variables, and Stochastic Processes, third edition, Mc-GRAW-HILL International Editions, 1991.

[22] Leon-Garcia A., Probability and Random Processes for Electrical Engineering, second edition, Addison-Wesley Publishing Company, 1992.

[23] Rowe H.E., Signals and Noise in Communication Systems, Van Nostrand Reinhold Company, 1965.

[24]. P. Faltstrom, M. Mealling, The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), IETF Network Working Group, RFC 3761, April 2004.

[25] <http://www.dnssec.org.mx>

[26] <http://brew.qualcomm.com/brew/en/>

[27] Carsten Schiefner, ENUM in Germany – 9.4.e164.arpa becoming commercial SWITCH ENUM Information Day 2006, DeNIC, June 29th, 2006

[28] JOSÉ IVÁN RAMÍREZ CARREÑO, Impacto en el desempeño de servidores recursivos que implementan DNSSEC., 2006.

[29] www.enum.org.mx

[30] <http://www.ntp.org/>

Tecnológico de Monterrey, Campus Monterrey



3002006934285

<http://biblioteca.mty.itesm.mx>