

THE QUALITY OF VOICE OVER IP

Tomi Yletyinen

**Helsinki University of Technology
Faculty of Electrical and Communications Engineering
Laboratory of Telecommunications Technology**

January 1998

Tomi Yletyinen

The Quality of Voice over IP

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in
Engineering:

Espoo, 27th of January, 1998

Supervisor:
Instructor:

Professor Raimo Kantola
M.Sc. Marko Luoma

Tekijä: Tomi Yletyinen	
Työn nimi: IP-puheen laatu	
Päivämäärä: 27 01.1998	Sivumäärä:
Osasto: Sähkö- ja tietoliikennetekniikan osasto	
Professori: S-38 Teletekniikka	
Valvoja: Professori Raimo Kantola	
Ohjaaja: DI Marko Luoma	
<p>Tässä työssä käydään läpi laatu kysymyksiä liittyen puheen siirtoon pakettiverkoissa käyttäen Internet Protokollaa (IP). Erityisesti keskitytään kahteen laatuparametriin: puhepakettien viiveeseen ja viiveen vaihteluun. Näitä tutkitaan mittauksin ethernetissä, perinteisessä reitittävässä verkossa sekä IP-kytketyssä verkossa.</p> <p>Puheen viivemittauksissa tutkitaan kokonaisviivettä lähettävän päätelaitteen mikrofonin ja vastaanottavan päätelaitteen kuulokkeen välillä. Mittauksissa käy ilmi, että päätelaitteiden aiheuttamat viiveet ovat merkittävä tekijä. Verkkojen ollessa järkevästi toteutettuja suurin osa viiveestä aiheutuu prosessoimista päätelaitteesta.</p> <p>Työssä tutkitaan myös tasa välein lähetettyjen puhepakettiparien saapumisvälien ja lähetysvälien eron vaihtelua. Yhteyden ollessa hyvä välien pakettien saapuessa tulisi olla sama kuin lähetyksessä. Mittauksissa selviää, että reitittävän ja kytkevän IP- verkon välillä on eroa tässä suhteessa. IP-kytkentää käyttävä verkko on hieman parempi kuin reititystä käyttävä verkko.</p>	
Avainsanat:	ATM, B-ISDN, IP-kytkentä, internet, liikennemittaukset, puheen siirto Internetissä, VoIP, puheen siirto pakettiverkossa, Voice over IP

Author:	Tomi Yletyinen	
Title:	The Quality of Voice over IP	
Date:	27.01.1998	Number of pages:
Faculty:	Electrical and Communications Engineering	
Professorship:	S-38 Telecommunications Technology	
Supervisor:	Professor Raimo Kantola	
Instructor:	M.Sc. Marko Luoma	
<p>This thesis examines the quality issues related to the transmission of voice over packet networks using the Internet Protocol (IP). The two quality parameters we concentrate on are delay and delay variance. Through measurements we examine these parameters in an ethernet, an IP forwarding- and an IP switching network.</p> <p>We measure the end-to-end delay from the microphone of the sending terminal to the speaker of the receiving terminal. The measurements indicate that the processing in the terminal is a major factor of the end-to-end delay. When the networks are implemented with sufficient resources the end-to-end delay is mainly caused by the terminal processing delay</p>		
<p>We also compare IP forwarding and IP switching networks in terms of how much interarrival jitter they cause to the speech packets. We measure the packet spacing differences at the receiver compared to that at the sender for speech packet pairs. The measurements indicate that the IP switching network causes a little less interarrival jitter than the IP forwarding network.</p>		
Keywords:	ATM, B-ISDN, IP-switching, internet, traffic measurements, Voice over IP, Voice over packet networks, Voice traffic measurements	

PREFACE

The work on this thesis was carried out at the Laboratory of Telecommunications Technology, Helsinki University of Technology as part of the Tekes funded IPANA-project. The industrial partners supporting the project are Helsingin Puhelin, Nokia Research Center, Nokia Telecommunications, Telecom Finland and Tellabs.

Cities are not built over night and a thesis seems to take around a year to finish. I started work on this thesis, at first half time December 1996 and full time from February 1997 to the maybe not so bitter end in December. This thesis concludes my studies for the degree of Master's of Science in Engineering, and so it was a huge leap for me, but probably not to the man kind. However small the progress, I did not do it alone. I want to thank my supervisor professor Raimo Kantola for his inspiring guidance and support all throughout my work.

Many thanks to Marko Luoma, my instructor for the fruitful discussions in getting my work on the right track, his help in configuring the networks and the analyzers and for his constructive criticism. I also thank Markus Peuhkuri for his cooperation in the configuration of the measurement platforms and the research network, for the help in planning of the measurements and for writing the fabulous client-server script.

I want to thank all my colleagues in the VoIP project group, and IPANA-project team. Special thanks to Pekka Lahtinen from Nokia Research Center, the man with the insight on voice over IP and also one of the fathers of the project. I also want to thank Pekka Pessi, also from Nokia Research Center for giving his opinions in the early stages of my work, and for aiding me in getting some of the needed material.

My sincere thanks to professor and head of the lab Timo Laakso for creating a pleasant working environment. Many thanks to my good friend Stefan Werner for helping me out with the adaptive algorithms. Special thanks to Stefan, David, Jose, Ramin, Jukka and Pasi, for keeping my spirits up and for making work fun at the lab.

Last, but not least my thanks and love to Marika for being there.

Tomi Yletyinen

Espoo, December 1997

CONTENTS

PREFACE	V
CONTENTS	VII
TABLE OF FIGURES	IX
LIST OF TABLES	X
LIST OF ABBREVIATIONS	XIII
1. INTRODUCTION	1
1.1. MOTIVATION.....	1
1.2. GOALS OF THE THESIS.....	2
1.3. STRUCTURE OF THESIS.....	2
2. VOICE OVER IP CONCEPTS	4
2.1. WHAT VOICE OVER IP	4
2.2. WHY VOICE OVER IP	4
2.3. APPLICATIONS OF VOICE OVER IP	4
2.4. THE IP PHONE TERMINAL EQUIPMENT	5
2.5. IP VOICE GATEWAY	6
2.6. IP VOICE RELATED STANDARDS AND PROTOCOLS.....	6
2.6.1. <i>The Internet Protocol - IPv4</i>	6
2.6.2. <i>The Next Generation of IP - IPv6</i>	7
2.6.3. <i>Transport Control Protocol</i>	8
2.6.4. <i>User Datagram Protocol</i>	8
2.6.5. <i>IETF Realtime Protocol</i>	9
2.6.6. <i>H.323</i>	12
2.6.7. <i>IP Switching</i>	15
2.7. SPEECH CODING	18
2.7.1. <i>Speech and Audio Coders for VoIP and their Quality</i>	18
2.8. VOICE SOURCE CHARACTERIZATION	19
2.8.1. <i>Packetization Process</i>	20
3. VOIP QUALITY OF SERVICE ISSUES	22
3.1. WHAT IS QoS	22
3.2. QoS CHARACTERISTICS	22
3.2.1. <i>End-to-end Transfer Delay</i>	22
3.2.2. <i>Throughput</i>	24
3.2.3. <i>Packet Loss</i>	25
3.3. QoS MANAGEMENT.....	25
3.3.1. <i>Use of RTCP in Measuring QoS</i>	26
3.3.2. <i>Procedures for maintaining QoS</i>	26
3.4. QoS AGREEMENTS	26
3.4.1. <i>The Internet Integrated Services Architecture</i>	27
4. SYNCHRONIZATION	29
4.1. SYNCHRONIZATION CONCEPTS	30
4.2. ADAPTIVE PLYOUT DELAY ESTIMATION	31
4.3. SYNCHRONIZATION QUALITY OF SERVICE.....	31
5. RELATED WORK	34
5.1. INTERNET END-TO-END MEASUREMENTS.....	34
5.2. REDUNDANT AUDIO AND DYNAMIC QoS CONTROL	35
6. MEASUREMENTS AND RESULTS	36
6.1. BACKGROUND.....	36
6.1.1. <i>The Goals of the Measurements</i>	36
6.1.2. <i>Choosing the Measurement Environments</i>	36
6.2. CALIBRATION MEASUREMENT - VOIP OVER ETHERNET	37
6.2.1. <i>Setup of the Measurement</i>	37
6.2.2. <i>Measurement of the total delay</i>	37
6.2.3. <i>End-to-End Delay of PC</i>	38
6.2.4. <i>Measurement of the Delay Variance of the Voice Packets</i>	39

6.2.5. <i>Analysis and Conclusions of the Ethernet Measurements</i>	40
6.3. MEASUREMENT II: VOIP OVER PACKET FORWARDING NETWORK	44
6.3.1. <i>Setup of the Measurement</i>	44
6.3.2. <i>Measurement of the Voice Packet Spacing Differences</i>	45
6.4. MEASUREMENT III : VOIP OVER IP SWITCHING NETWORK	47
6.4.1. <i>The Setup of the Measurements</i>	47
6.4.2. <i>Measurement of the Delay Variance of the Voice Packets</i>	47
6.4.3. <i>Packet Spacing Differences and Interarrival Jitter Estimates</i>	48
7. ANALYSIS	56
7.1. COMPARISON OF IP FORWARDING AND IP SWITCHING	56
7.1.1. <i>D frequencies</i>	56
7.1.2. <i>D Percentiles</i>	59
7.2. MEASUREMENT METHODS	62
7.3. SUMMARY	62
CONCLUSIONS AND FUTURE WORK	64
APPENDIX A: SOME MORE FIGURES	66
APPENDIX B: IP SWITCHING WITH PRIORITIES	70
REFERENCES	73

TABLE OF FIGURES

Figure 2-1: The framework for IP voice.....	5
Figure 2-2: Typical IP voice terminal with video conferencing capabilities.	5
Figure 2-3: IP Voice to PSTN gateway.....	6
Figure 2-4: The Voice over IP protocol stack	6
Figure 2-5: IPv4 header.....	7
Figure 2-6: IPv6 header.....	8
Figure 2-7: TCP header.....	8
Figure 2-8: UDP header	8
Figure 2-9: RTP in relation to the protocol stacks.	9
Figure 2-10: RTP header []	10
Figure 2-11: Sender report RTCP [7].....	11
Figure 2-12: Receiver report [7].	12
Figure 2-13: H.323 terminal and H.323 scope.	12
Figure 2-14: H.323 entities.....	13
Figure 2-15: Simple call setup signaling.	14
Figure 2-16:The scope of H.225.0.	14
Figure 2-17: The structure of IP-switches and the protocols.....	15
Figure 2-18: The Ipsilon protocol hierarchy.	16
Figure 2-19: The IP switch as a packet forwarding router.	16
Figure 2-20: Soft-state routing anf flow control.	16
Figure 2-21: A switched flow in IP switching.	17
Figure 2-22: IP switching network.	17
Figure 2-23: The traffic characteristics of a voice stream.	20
Figure 2-24: Two-state model of a voice source.	20
Figure 2-25: From voice to bit stream and back.....	21
Figure 3-1: IETF traffic service class hierarchy.	27
Figure 4-1: Adaptive synchronization.	30
Figure 6-1: Setup of the VoIP over ethernet measurement.....	37
Figure 6-2: Packet spacing difference distribution (frequencies) of VoIP over non-loaded Ethernet.	41
Figure 6-3: Packet spacing difference percentiles of VoIP over non-loaded Ethernet.	41
Figure 6-4: Packet spacing difference distribution (frequencies) of VoIP over Ethernet loaded with small packets.	42
Figure 6-5: Packet spacing difference percentiles of VoIP over Ethernet loaded with small packets.	42
Figure 6-6: Packet spacing difference frequencies of VoIP over Ethernet with a bursty load.	43
Figure 6-7: Packet spacing difference frequencies of VoIP over Ethernet with a bursty load (different scaling).	43
Figure 6-8: Packet spacing difference percentiles of VoIP over Ethernet with a bursty load.....	43
Figure 6-9: Packet spacing difference frequencies of VoIP over Ethernet loaded with large packets.....	44
Figure 6-10: Packet spacing difference percentiles of VoIP over Ethernet loaded with large packets.....	44
Figure 6-11: Setup of the VoIP over IP forwarding measurement.	45
Figure 6-12: Network traffic during measurement K-S 8.....	46
Figure 6-13: Network traffic during measurement K-S 2.....	46
Figure 6-14: Setup of the IP switching measurement.	47
Figure 6-15: Network traffic during measurement K-S 2.....	48
Figure 6-16: Packet spacing differences of IP forwarding with flood ping load.	49
Figure 6-17: Packet spacing differences of IP switching with flood ping load.	49
Figure 6-18: Packet spacing differences of IP forwarding with K-S 2 processes.....	50
Figure 6-19: Packet spacing differences of IP switching with K-S 2 processes.	50
Figure 6-20: Packet spacing differences of IP forwarding with K-S 4 processes.....	51
Figure 6-21: Packet spacing differences of IP switching with K-S 4 processes.	51
Figure 6-22: Packet spacing differences of IP switching with mixed load.....	52
Figure 6-23: Percentiles of errors in the estimates of J	52
Figure 6-24: Percentiles of errors in the estimates of J	53
Figure 6-25: Percentiles of errors in the estimate of J.....	53
Figure 7-1: <i>D</i> frequencies of IP forwarding with no load.....	56
Figure 7-2: <i>D</i> frequencies of IP switching with no load.....	56
Figure 7-3: <i>D</i> frequencies of IP forwarding with flood ping load.	57

Figure 7-4: <i>D</i> frequencies of IP switching with flood ping load.	57
Figure 7-5: <i>D</i> frequencies of IP forwarding with K-S 2 processes.....	58
Figure 7-6: <i>D</i> frequencies of IP switching with K-S 2 processes.....	58
Figure 7-7: <i>D</i> frequencies of IP switching with mixed load.....	59
Figure 7-8: <i>D</i> percentiles of IP forwarding with no load (% <i>D</i> < <i>n</i> ms).....	59
Figure 7-9: <i>D</i> percentiles of IP switching with no load (% <i>D</i> < <i>n</i> ms).	60
Figure 7-10: <i>D</i> percentiles of IP forwarding with flood ping load.	60
Figure 7-11: <i>D</i> percentiles of IP switching with flood ping load.	60
Figure 7-12: <i>D</i> percentiles of IP forwarding with K-S 2 processes.....	61
Figure 7-13: <i>D</i> percentiles of IP switching with K-S 2 processes.....	61
Figure 7-14: <i>D</i> percentiles of IP switching with mixed load.....	61
Figure 7-15: <i>D</i> frequencies of IP forwarding with K-S 4 processes.....	66
Figure 7-16: <i>D</i> frequencies of IP switching with K-S 4 processes.....	66
Figure 7-17: <i>D</i> frequencies of IP forwarding with K-S 8 processes.....	67
Figure 7-18: <i>D</i> frequencies of IP switching with K-S 8 processes.....	67
Figure 7-19: <i>D</i> percentiles of IP forwarding with K-S 4 processes.....	67
Figure 7-20: <i>D</i> percentiles of IP switching with K-S 4 processes.....	68
Figure 7-21: <i>D</i> percentiles of IP forwarding with K-S 8 processes.....	68
Figure 7-22: <i>D</i> percentiles of IP switching with K-S 8 processes.....	68
Figure 7-23: Packet spacing differences of IP forwarding with K-S 8 processes.....	69
Figure 7-24: Frequencies of <i>D</i> of the voice stream.....	70
Figure 7-25: Percentiles of <i>D</i> of the voice stream.....	70
Figure 7-26: Frequencies of <i>D</i>	71
Figure 7-27: Percentiles of <i>D</i>	71

LIST OF TABLES

Table 2-1:Type of Service in IP	7
Table 2-2:Flow types.....	17
Table 2-3:Speech band coder comparison.....	19
Table 3-1: Required bandwidths if no header compression is used.....	24
Table 4-1:Quality of Service for Synchronization Purposes.	31
Table 6-1:Terminal delays.	38
Table 6-2:End-to-end delays over Ethernet.....	38
Table 6-3:The PC details.....	38
Table 6-4: Terminal and end-to-end delays.....	38
Table 6-5: VoIP over Ethernet with no background load.....	40
Table 6-6:Traffic summary.	40
Table 6-7:VoIP over Ethernet with different background loads.....	40
Table 6-8:Summary of IP forwarding measurements.	45
Table 6-9:IP forwarding measurements traffic summary.	46
Table 6-10:Summary of IP switching measurements	47
Table 6-11:IP switching measurements traffic summary.....	48
Table 6-12:A summary of the maximum errors of the estimates.....	54
Table 6-13:A summary of the negative maximum errors of the estimates.	54

LIST OF ABBREVIATIONS

A

ABR	Available Bit Rate
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMARP	ATM Address Resolution Protocol

B

BGP	Border Gateway Protocol
-----	-------------------------

C

CBR	Constant Bit Rate
CELP	Code Exited Linear Prediction

D

DSI	Digital Speech Interpolator
DSP	Digital Signal Processor

F

FDDI	Fiber Distributed Data Interface
FF	Fixed-Filter, style of reservations in RSVP
FlowSpec	Flow Specification

G

GSM	Global System for Mobile communications
-----	-----------------------------------------

H

HS	Hard State
----	------------

I

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IETF	Internet Engineering Task Force
IGMP	Internet Group Multicast Protocol
IGP	Interior Gateway Protocol
IIS	Internet Integrated Services
IMA	International Multimedia Association
IMTC	International Multimedia Teleconferencing Consortium
IP	Internet Protocol
IPANA	IP and Atm advanced Network Architectures
IS	Integrated Services
ISA	Integrated Services Architecture
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITCA	International Teleconferencing Association
ITIC	Internet Telephony Interoperability Consortium
ITU	International Telecommunications Union
ITU-T	International Telecommunications Union

L

LAN	Local Area Network
LAN	Local Area Network
LANE	LAN Emulation
LDAP	Lightweight Directory Access Protocol
LDU	Logical Data Unit
LIS	Logical IP Subnetworks

M

MCS	Multipoint Communication Services
MC	Multipoint Controller
MCU	Multipoint Control Unit
MIT	Massachusetts Institute of Technology
MOS	Mean Opinion Score
MPOA	Multi Protocol Over Atm

MSTP Multimedia Synchronization Transport Protocol.

N

NAP Network Access Point
nrtVBR Non-realtime Variable Bit Rate

P

PARC Xerox Palo Alto Research Center
PDU Protocol Data Unit
PSDN Packet Switched Digital Networks
PSQM Perceptual Speech Quality Measure-
PSTN Public Switched Telephone Network
PVC Permanent Virtual Circuit

Q

QoS Quality of Service

R

RARP Reverse ARP
RCAP connection management protocol in Tenet Suite
RFC Request For Comments
RMTSP Realtime Message Transport Protocol (Tenet Suite)
Rspec A flow specification parameter specifying the QoS desired
RSVP resource ReSerVation Protocol
RTCP Real Time transport Control Protocol
RTCP RTP control protocol
RTP Real Time Protocol
RTP Realtime Transport Protocol
rtVBR Realtime Variable Bit Rate

S

SE Shared-Explicit, style of reservations in RSVP

T

TASI Time-Assigned Speech Interpolator
TCP Transmission Control Protocol
TOS Type of Service
Tspec A parameter specifying the flow type

U

UBR Unspecified Bit Rate
UDP User Datagram Protocol
ULS User Location Service
UNI User to Network Interface

V/ W

WAN Wide Area Network
VC Virtual Circuit
WF Wildcard-Filter, style of reservations in RSVP
WFQ Weighted Fair Queuing
VoIP Voice over IP
VP Virtual Path

1. INTRODUCTION

The evolution of workstations and personal computers from simple word processors to number crunchers with multimedia capabilities, the increased connectivity to high speed networks and the advances in signal processing and programming have facilitated a new family of distributed realtime multimedia applications. Distributed multiparty audio and video conferencing has gained wide-spread acceptance, and seems to be one of the applications driving the concept of a network that integrates all services.

For a long time ATM was seen as the network that will integrate all the new services, and the general point of view was that current data and telephone networks are just single service networks. This has proven wrong to some extent. The Internet of today is used for a number of realtime applications such as wide area audio and video conferencing, distributed gaming and audio and video broadcasting. Realtime applications are sensitive to the time of arrival of the transmitted data. The temporal relationships should be the same at the sender and receiver. The network should not impose too much delay, the delay should be constant and the packets should arrive as close as possible in-order. The value of the data to receiver is dependent on the fulfillment of these requirements.

Internet is still based on the internet protocol developed in the seventies and is so far delivering only best-effort service. This means that the quality a user of realtime applications sees is sometimes reasonable, but at times of high network congestion intolerable. Packet losses as high as 10 - 30% are not uncommon in the Internet of today. The application developers have tried to design their applications tolerant and adaptive to changes in the network conditions. The behavior of networks under different loads has been studied through measurements and this knowledge has been applied to the design of the applications. New coding schemes, which add redundancy to the transmitted data and through error correction reproduce lost packets are emerging along with improved ways to estimate delay and delay variance.

New technologies are being developed that try to provide something better than the traditional best-effort service to the Internet world. One of these is the concept of cut-through switching with flow detection used by various IP switching technologies.

This thesis concentrates on the quality issues of one special class of realtime applications, Internet telephony or voice transmission over IP. Through an extensive literature study and measurements we will examine the delay and delay variance behavior of IP Voice in a routing network and compare the results with those of an IP switching network, and to compare the interarrival jitter estimators of the applications.

1.1. Motivation

The transmission of voice over packet networks was an active research area in the late 70's and early 80's. At that time the research concentrated in the basic concepts, such as the performance evaluation of multiplexing voice packets using statistical multiplexing and silence detection compared to using circuit switching with time division multiplexing. Enormous advances have been made after those times in computer technology, and it is now possible to use packetized voice over LANs and WANs.

The current networks using Internet Protocol are said to have quality problems in transferring realtime voice, e.g.: delay, delay variance and packet loss are parameters that need improvement. Voice over IP is again a hot topic in research. The problems of packetized realtime voice are solved in two fronts: 1) new technologies in making applications cope in best-effort Internet 2) improving networks.

The questions we want to answer are :

1. What are the characteristics of the quality of Voice over IP ?
2. What are delay and delay variance characteristics of Voice over IP in an IP switching environment?
3. Is the quality of Voice over IP over IP switching better than over IP forwarding, if we compare them with objective parameters such as delay variance?
4. How good are the delay variance estimators designed for best-effort Internet in a campus IP-switching network/IP forwarding network/Ethernet?

1.2. Goals of the Thesis

The purpose of this thesis is to serve as an introduction to the quality issues of voice over IP. The object is to try to shed light on the QoS characteristics of voice over IP. The main impairments of VoIP are the delay and packet loss. Our focus is in LANs and campus networks and the future Internet and intranets implemented with adequate resources. Through measurements we will try to understand the delay behavior of IP Voice in a campus network implemented with IP forwarding network and the same network implemented with IP switching. We will also analyze the performance of the estimators for delay and delay variance in these networks.

1.3. Structure of Thesis

Chapter two begins with an introduction to the main concepts related to Voice over IP: it's applications, the protocols and the standards. In the same chapter some of the fundamentals of packet voice are presented: voice coding, voice traffic models and the voice packetization process. Chapter three introduces the quality issues related to Voice over IP: the characteristics of the QoS of VoIP, the management of QoS and the QoS agreements of VoIP. Chapter four explains the synchronization concepts and goes through the mathematics behind playout delay estimation. Chapter five introduces the related work. In chapter six the measurements are explained and the results of the measurements are give. In chapter seven the results are analyzed and compared.

2. VOICE OVER IP CONCEPTS

Voice over IP (VoIP) is the transmission of voice over networks using the Internet Protocol. IP-networks have become increasingly popular in the past few years, the exponential growth of the public Internet leading the way in to the IP-world.

In this chapter the basic concepts relating to voice over IP will be presented: what is Voice over IP, what it is used for and why, the applications, the main standards and protocols relating to VoIP: IP, UDP, TCP, RTP, H.323 and IP Switching. In the end of the chapter we will present the basics of packetized voice: coding and voice source modeling.

2.1. What Voice over IP

The transmission of voice signals over packet networks is not a new concept. It was an active research topic in the late 70's and early 80's. The feasibility factors are obvious: a speech source is active only half of the time and thus it makes sense to use speech activity detection, only send when a source is active and multiplex a number of sources on one link either relying on statistical multiplexing or employing some kind of time slotted system (TSM). Already then, in the late 70's there was discussion and even experiments with packetized voice over the ARPANET, the predecessor of the Internet using IP and specialized coding and packetizing equipment [1]. The conclusion then was that packetized voice has economical advantages and can be done. Still it took some fifteen years for the packetized voice to gain popularity. The specialized equipment is no longer required: a desktop general use personal computer, a sound card, microphone and speakers and some software are all that is needed. The equipment is already built into most multimedia capable computers today, and with the wide-spread connectivity to the internet local and wide area telephony over packet data networks is possible for a very large audience.

2.2. Why Voice over IP

At first the transmission of voice over data networks seems like a bad idea. We already have a well functioning circuit switched phone network that extends through out the seven continents, and forms the largest machine ever built by man. Data networks on the other hand are presently ill-suited for the transmission of voice. Voice is realtime and requires realtime handling from the network. Most data networks currently do not provide realtime service. Still IP-voice has found a market, the main driver being surpassing the costly long-distance telephony. The popularity of virtually free long-distance calls have proven the fact that even poor quality is satisfactory if the price is right, and that the old American sales strategic motto, "In a competitive and developed market three things are important: price, price and price", has proven right again.

In the future long-distance call charges will sink, not because of VoIP, but because of the increasing competition. The cost advantage of VoIP will diminish, but still market experts are predicting a bright future for it. Because of statistical multiplexing and advanced compression methods VoIP should in principal be more cost effective than circuit switched voice transmission. Video conferencing and CSCW (Computer supported collaborative work) applications are the new drivers of VoIP.

2.3. Applications of Voice over IP

There are three basic network scenarios for Voice over IP:

- Computer to Computer
This is the basic scenario: both the A and B subscribers are using computers attached to an IP network as terminals.
- Computer to Phone and Phone to Computer
In this scenario one of the subscriber is using a computer for IP-voice and the other uses a phone on a PSTN/ISDN/GSM/TDM network. A gateway on the edge of the IP network translates IP-voice to voice and takes care of the signaling between the two networks.
- Phone to Phone
Both subscribers are using conventional phones in this option, and the IP network is used for the long distance connection. Gateways on both ends take care of translations between networks.

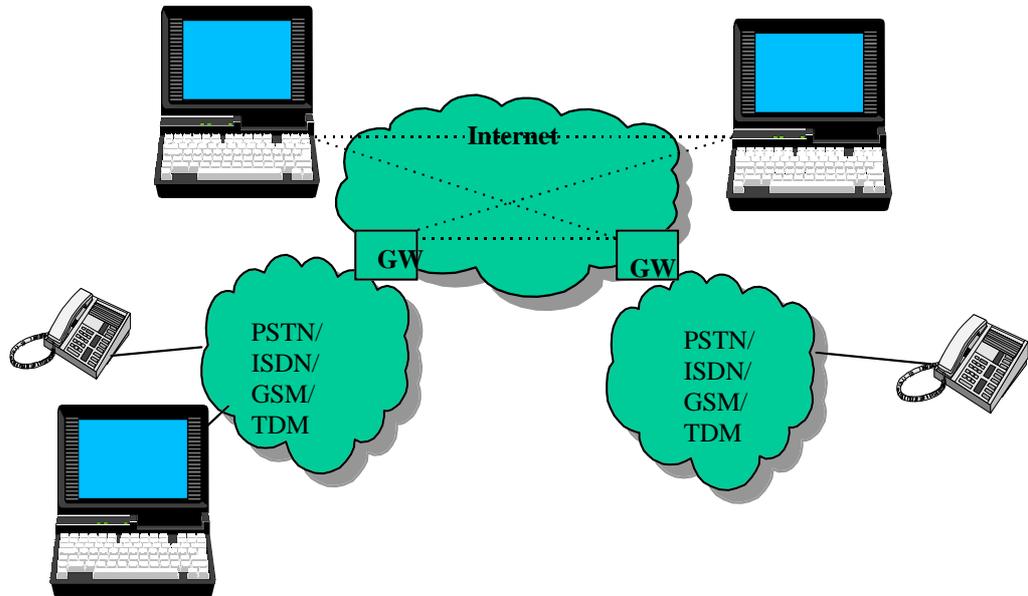


Figure 2-1: The framework for IP voice

2.4. The IP Phone Terminal Equipment

The IP phone terminal equipment can be workstations or personal computers equipped with IP phone software, sound card, speakers, a microphone and of course a network interface. The first experimental terminal software was designed for UNIX workstations with plenty of processing capability and memory, but the Pentium equipped PCs of today can do the job as well. The minimum requirements are mostly stated as the 486 class, but in our experiments a 100 MHz Pentium was barely sufficient. If video is used with audio, then the latest Pentium II MMX processor equipped computer could be barely sufficient. An ordinary telephone handset gives more comfort and better audio quality than the speaker/microphone combination.

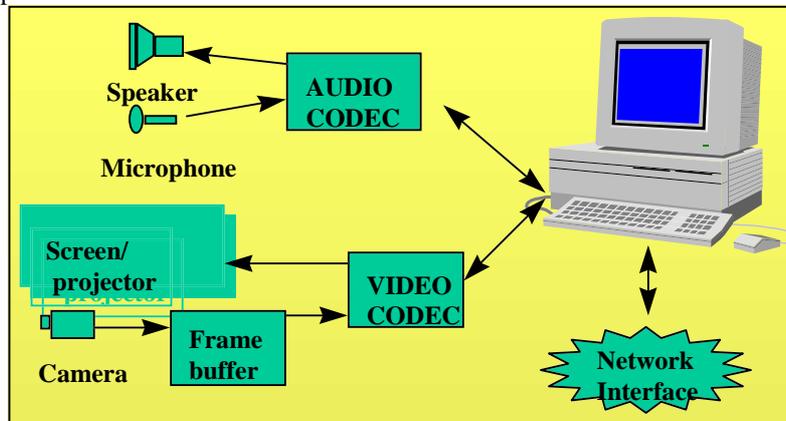


Figure 2-2: Typical IP voice terminal with video conferencing capabilities.

2.5. IP Voice Gateway

An IP voice gateway is an interworking unit capable of translating IP-phone signals to ordinary telephone signals, capable of IP address to telephone number conversion and foremost can signal open a connection from the IP network to a terminal in the telephone network. So far the gateways have been built around pc hardware. The processing intensive coding and decoding of voice is done with special DSP-equipped telephony boards. The control functions are run in main memory using the CPU. There are scaling and reliability problems in an architecture like this, but the cost is also relatively low compared to for example a PBX.

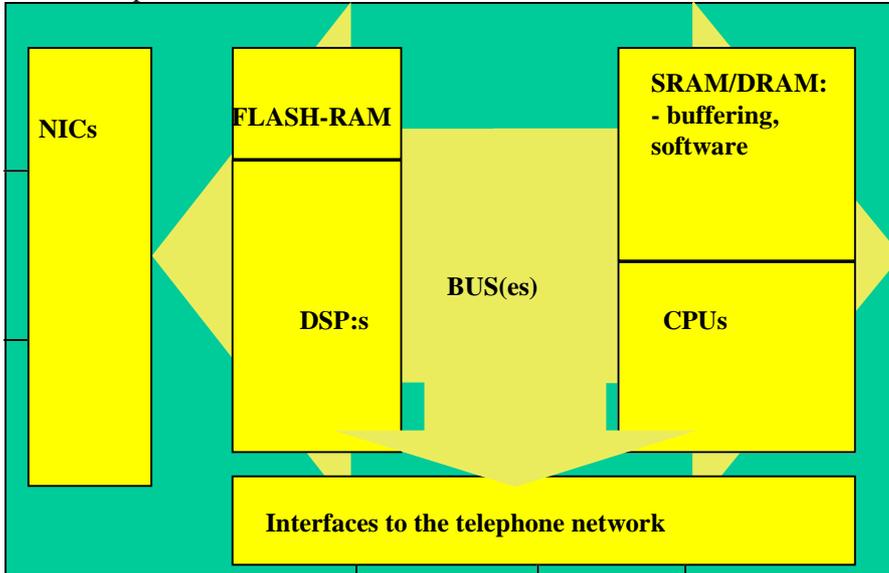


Figure 2-3: IP Voice to PSTN gateway

2.6. IP Voice Related Standards and Protocols

In this section we present the most important protocols and standards related to Voice over IP. We will start with Internet Protocol: the handling of types of services in IP networks, fragmentation and reassembly and IP options.

This will be continued by an introduction to UDP and TCP protocols, the two transport layer protocols of IP. UDP is mostly used for realtime transmission of voice and TCP in relation to VoIP for streaming control and applications where added buffering delay is acceptable, e.g. audio broadcast. To take use of TCP retransmissions, buffering is needed. Waiting for retransmissions is not appropriate for realtime communications, where interactivity needs to be maintained. RTP is the session layer protocol used for synchronization, multicast session participant information relaying and recipient network quality monitoring purposes.

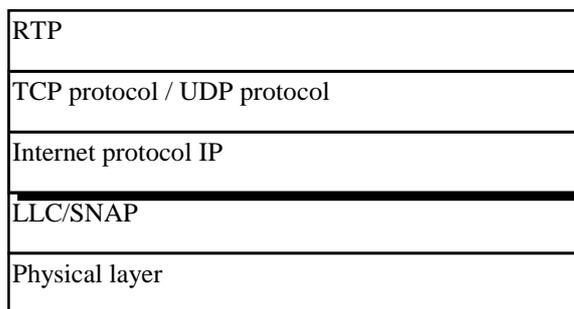


Figure 2-4: The Voice over IP protocol stack

2.6.1. The Internet Protocol - IPv4

RFC-791 [2] defines the latest revised version of U.S. Department of Defense Standard Internet Protocol, better known as IPv4. The internet protocol is used for interconnected systems of packet-switched computer communication networks. The services internet protocol provides for are: transmitting blocks of data called datagrams from source to destination and fragmentation and reassembly of long datagrams, if necessary.

The IP header carries Internet source and destination addresses as well as a number of parameters needed for the routing of packets. In Figure 2-5 is the IPv4 header [2]. As can be seen the header comprises of fixed fields present in every IP packet, and several options. Each row in the header is 32 bits long and the bits are transmitted with most significant bit first. The first field in the header is the version field, used for distinguishing between other “IP compatible” protocols and previous versions of IP. The current IP version number is four. The time to live (TTL) field was initially an indication of the maximum lifetime of the packet in the network in seconds. IP multicast uses the TTL field to set the scope of the session. The multicast traffic is not sent outside of the site if the TTL is 15 or lower. With TTL of 63 the session will cover a region, and session with TTL of 127 or larger will cover the world.

Ver	IHL	ToS	Total length	
Identification			Flags	Fragment offset
Time to live	Protocol		Header Checksum	
Source Address				
Destination Address				
Options			Padding	

Figure 2-5: IPv4 header

The second octet of the packet header defines the “type of service”, ToS. It is divided in two fields, the “precedence” and the “type of service”. Precedence is an indication of the priority and the “type of service” is an indication of routing. The routing protocols are supposed to compute a default route when no ToS bit is set, a shortest route (D set), a largest throughput route (T set), a most reliable route (R set) and a cheapest route (C set). The bits should not be normally combined. In general the ToS is not used in the Internet because of fear of misuse, although the “type of service” routing is defined in some routing protocols (BGP and OSPF) and some routers are capable of priority queuing (e.g. Cisco routers). A network manager may choose to use ToS within a private network [3].

Table 2-1: Type of Service in IP

Parameter and length	Options
Precedence (3 bit)	8 values
D, Delay (1 bit)	Low (0) and high (1)
T, Throughput (1 bit)	Low (0) and high (1)
R, Reliable route (1 bit)	Low (0) and high (1)
C, Cheapest route (1 bit)	Low (0) and high (1)

2.6.2. The Next Generation of IP - IPv6

IP next generation (IPng) [4], [5] is a new version of the Internet Protocol designed as a successor to the IP version 4. The version number assigned for IPng is 6 and it is formally called IPv6. It was not designed to be a giant leap away from IPv4 - many of functions of IPv4 were kept. The primary motivation for the design of IPng was the exponential growth of the Internet leading to running out of IP address space. IPng offers: expanded routing and addressing capabilities, auto-configuration of addresses, improved scalability of multicast routing by adding a "scope" field to multicast addresses, a simplified header format (despite the increased address size, headers have only doubled), improved support for options, quality-of-service capabilities (flow labeling), authentication and privacy capabilities. The IPng protocol consists of two parts, the basic IPng header and IPng extension headers.

Ver	Priority	Flow label	
Payload length		Next header	Hop limit
Source Address			
Destination Address			

Figure 2-6: IPv6 header

2.6.3. Transport Control Protocol

IP provides best-effort service. There is no guarantee that all datagrams arrive at the destination. Local congestion may cause discarding of queued packets, and transmission errors may cause packet loss. Packets may also be spread across several parallel routes and arrive in disorder. The transport control protocol, TCP was designed to hide these errors. A TCP connection is set between two ports. A TCP port is identified by an IP address and a 16-bit port number, which identifies an application within the host.

TCP is used as an alternative to user datagram protocol for VoIP. When a packet is lost (or delayed long enough) the sender re-transmits the packet. For a realtime application the re-transmitted packet is usually worthless, because the playout point of the packet (or samples in a packet) has already been missed. Therefore TCP is only appropriate in regard to VoIP for applications where the timely deliver is not required, e.g. signaling of a IP telephony connection, Internet audio broadcasts and audio on demand. In Figure 2-7 is depicted the TCP header.

Source Port		Destination Port	
Sequence Number			
Acknowledgement Number			
Offset	Reserved	Control	Window
Checksum		Urgent Pointer	
Options		Padding	
Data			

Figure 2-7: TCP header

2.6.4. User Datagram Protocol

The user datagram protocol is an alternative for TCP. Through UDP the applications get direct access to the datagram service of IP. Applications post packets to UDP ports identified by an IP address and a 16-bit port number. The UDP header (Figure 2-8) is lightweight and therefore appropriate for the transmission of short voice packets. As mentioned before realtime voice mainly uses UDP.

Source Port	Destination Port
Length	Checksum
Data octets	

Figure 2-8: UDP header

2.6.5. IETF Realtime Protocol

The IETF audio-video transport group started work on a realtime transport protocol in 1993. The aim of the protocol was at providing services required by interactive multimedia conferences, such as playout synchronization, demultiplexing, media identification and active-party identification. However, not only multimedia conferencing applications can benefit from RTP, but also storage of continuous data, interactive media distribution, distributed simulation, active batch, and control and measurement applications could take advantage of the possibilities RTP brings. [6]

The design goals of RTP were [6]:

1. Content flexible - RTP should not be limited to only voice and video conference;
2. Extensible - RTP should be able to accommodate new services as operational experience accumulates;
3. Independent of lower layer protocols - RTP should work with UDP, TCP, ST-II and ATM;
4. Bridge/RTP gateway compatible - it should be possible to aggregate several media streams into a single stream and possibly retransmit it with different encoding;
5. Bandwidth efficient - header overhead in short voice packets can be as much as 100%. For example with a 65ms packetization interval using 4800 bit/s encoding produces 39 byte packets. IPv4 incurs 20 bytes of headers, UDP an additional 8 bytes and the datalink layer at least an additional 8 bytes. With RTP headers around 4 to 8 bytes the total of headers is around 36 or 40 bytes per packet. This could stand in the way of running RTP over low-speed links;
6. International - a and μ law encoding as well as non US-ASCII character sets should be included;
7. Processing efficient - even the longest packetization intervals give packet arrival rates of 40 per second for a single voice channel. Per packet processing overhead may become a concern;
8. Implementable now - the protocol is more or less experimental and the lifetime of the protocol was not anticipated long, so it must be implementable with the current hardware and software.

2.6.5.1. The Architecture of RTP

RTP concept consists of two closely linked parts: the realtime transport protocol (RTP), for carrying data that has realtime properties and the RTP control protocol (RTCP), for monitoring the quality of service and conveying information about the participants in an on-going conference. RTP implementation will often be integrated into application processing rather than being implemented as a separate layer. The RTP framework is deliberately “loose” allowing for modifications and tailoring. In addition to RTP a complete specification for a particular application will require a payload format and a payload profile specification. A payload format defines how a particular payload (e.g. audio, video) is to be carried in RTP. A payload format specification defines how a set of payload type codes are mapped into payload formats (e.g. media encodings).

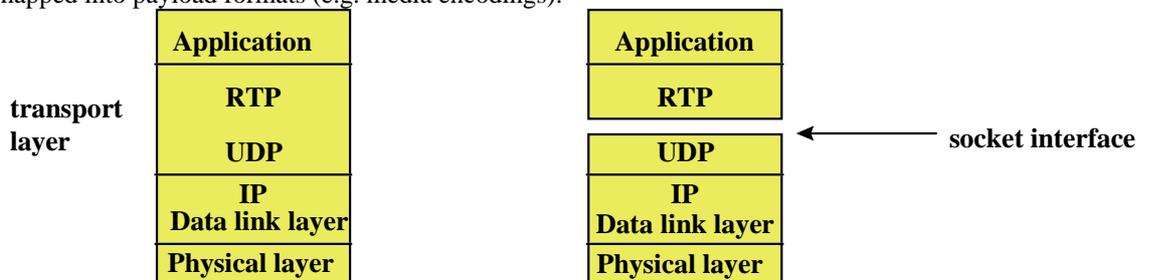


Figure 2-9: In applications RTP is typically run on top of UDP to make use of its port numbers and checksums. RTP can be viewed as a sublayer of transport layer. However, from the application developer's perspective, RTP is an integral part of the application. The developer must always integrate RTP into the application.

RTP-session setup consists of defining a pair of destination transport addresses one IP address plus a UDP port pair, one for RTP and one for RTCP. In the case of a multicast conference the IP address is a

class D multicast address. In a multimedia session each medium is carried in a separate RTP session, with its own RTCP packets reporting the quality of that session. Usually additional media are allocated in additional port pairs and only one multicast address is used for the conference.

2.6.5.2. RTP Packets

RTP smoothes out the effects of network delay variance e.g. performs synchronization, see [7]. This is done by adjusting the playout time so that the temporal relationships between samples are restored and late arriving packets are discarded. In order to do this the RTP header is added to the continuous media sample or a group of samples.

The RTP header format is in Figure 2-10. The fields included are:

- V - version information, for distinguishing between different versions of RTP (2 bits).
- P - padding (1 bit), if the padding bit is set the packet contains one or more additional padding octets at the end which are not part of the payload. Padding is needed by some encryption algorithms with fixed header sizes.
- X - extension, if the extension bit is set, the fixed header is followed by exactly one header extension.
- CC - CSRC count contains the number of CSRC identifiers that follow the fixed header.
- M - marker (1 bit), the marker bit is defined by the a payload profile. It is intended to allow significant events such as frame boundaries to be marked in a packet stream.
- PT - payload type (7 bits) identifies the format of the RTP payload and determines its interpretation by the application.
- Sequence number - (16 bits) increments by one for each RTP data packet sent. It may be used by the receiver for detecting packet loss and restoring packet sequence.
- Timestamp - (32 bits) a media specific timestamp containing the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and delay variance calculations.
- SSRC - (32 bits) synchronization source identifier identifies the synchronization source. The identifier is chosen randomly, so that no two sources within a session have the same SSRC identifier. The source identification is defined to identify a single timing and sequence number space. It is not sufficient to use the local network address (such as an IP address), because the address may not be unique. Also RTP is designed to be transport protocol independent. RTP translators are designed to provide the interoperability between multiple networks with different address space. Using a randomly chosen SSRC is simpler than using allocation patterns within two address spaces, and results in a much smaller probability of address collision. In case of the rare event that there are two sources in the same session with the same SSRC, collision and collision resolving are implemented.

CSRC list- (0-15 x 32 bits) the contribution source identifier identifies the contributing sources for the payload contained in the packet. CSRC identifiers are inserted in RTP mixers, which aggregate RTP packets from different sources into one RTP packet. In the case of audio packets, the SSRC identifiers of all sources that were mixed together to create the packet are listed in the CSRC. This is needed for correct talker indication at the receiver.

SSRC and CSRC are of course not relevant in a unicast session, if the source and the receiver can be identified using network protocol source field, such as in a two person call in the Internet, the source is identified by IP source address.

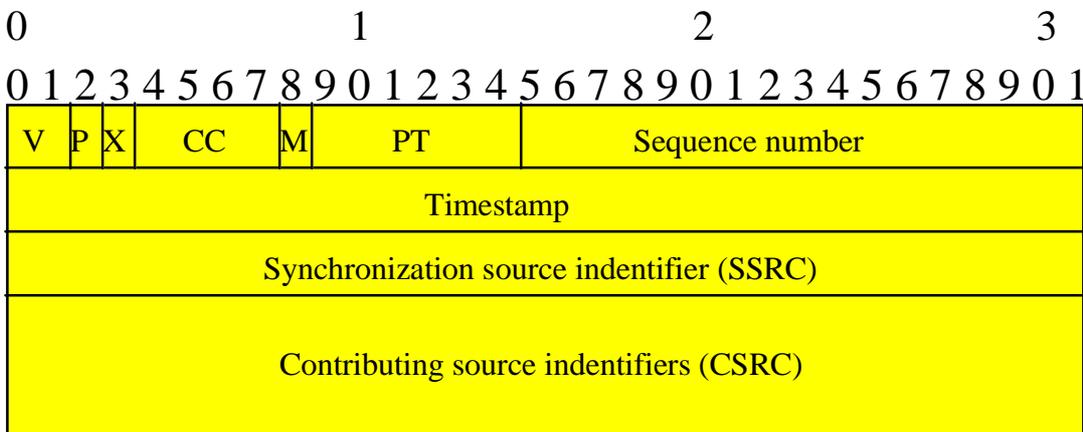


Figure 2-10: RTP header [7]

The use of RTP in removing delay variance from an audio stream transmitted over a packet network is in section 4.

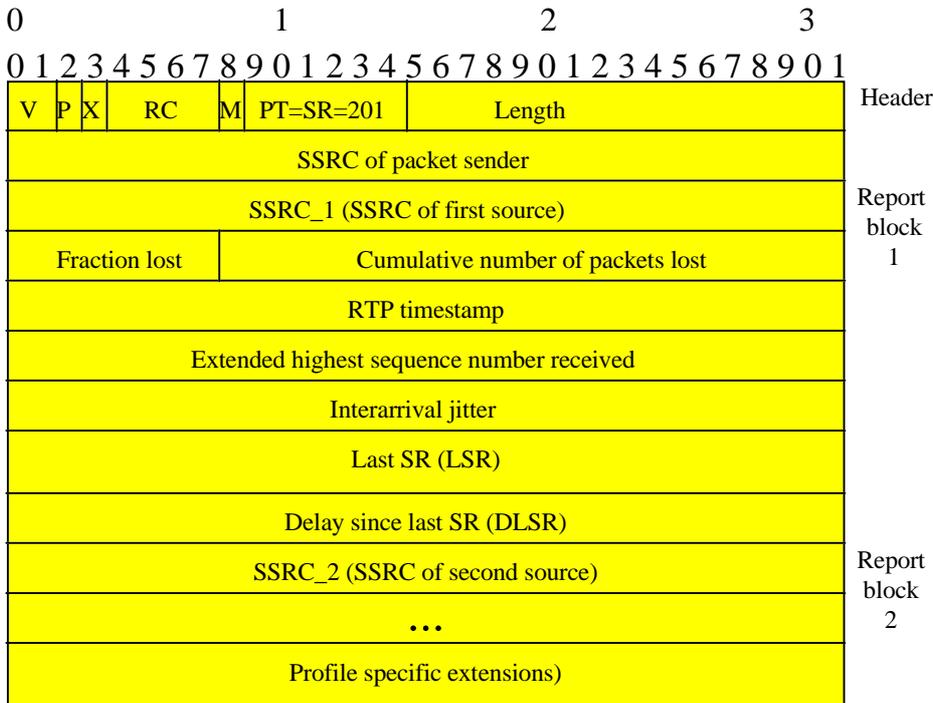


Figure 2-12: Receiver report [7].

2.6.6. H.323

ITU-T H.32x recommendations define visual telephone terminals and how they are run over various networks. H320 applies to N-ISDN while H.321 applies to B-ISDN (ATM). H.322 and H.323 apply to LANs. The difference between the latter two is in that H.323 applies to LANs without QoS guarantees and H.322 to LANs with QoS guarantees. H.323 is applicable to any packet-switched network regardless of the underlying physical layer. The network is expected to provide both an unreliable and a reliable delivery mechanism. In IP networks the former is Transmission Control Protocol, TCP and the latter User Datagram Protocol, UDP. H.323 is independent of the network topology: H.323 terminals can communicate through hubs, routers, bridges, and dial-up connections. In the scope of this thesis, delay and synchronization the recommendation H.225.0, part of H.323, is interesting in particular. Most of this section is from [8] The scope of recommendation H.323 is in Figure 2-13.

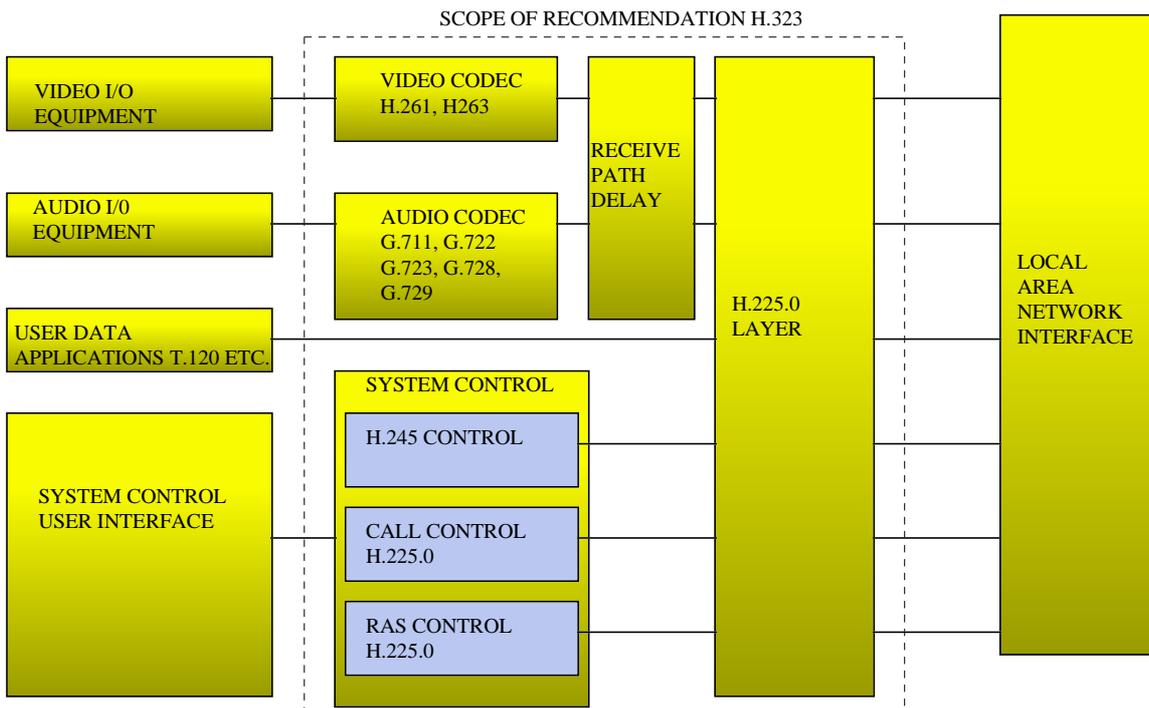


Figure 2-13: H.323 terminal and H.323 scope.

H.323 is an umbrella standard, in the sense that it is a collection of many ITU-T recommendations. H.323 provides system and control descriptions, as well as call model descriptions and call signaling procedures. H.225.0 describes media (audio and video) packetization, media stream synchronization, control message packetization, and control message formats. Recommendation H.245 describes the messages and procedures used for opening and closing logical channels for audio, video, data, camera mode requests, control, and indications. T.120 series of recommendations are used for data applications, such as shared whiteboard. For audio coding G.711 is mandatory, while G.722, G.728, G.723.1 and G.729 are optional. For video H.261 QCIF mode is mandatory, while H.261 CIF and H.263 modes are optional.

H.323 entities are:

- H.323 terminal, that provides realtime bi-directional audio, video and data communications;
- gatekeeper, that provides security functions and the means to control H.323 traffic on the net through admission control. It also provides address translation services, e.g. it converts telephone numbers to network addresses;
- multipoint Control Unit (MCU), needed in centralized and hybrid multipoint conferences. MCU is used for distribution of media streams. All terminals send their media streams to the MCU, which then distributes selected or mixed streams back to the terminals. In a decentralized multipoint conference each terminal distributes its media streams to all other terminals in the conference;
- gateway, provides translation of call signaling, control messages, and multiplexing techniques between H.323 and other ITU-T terminal types.

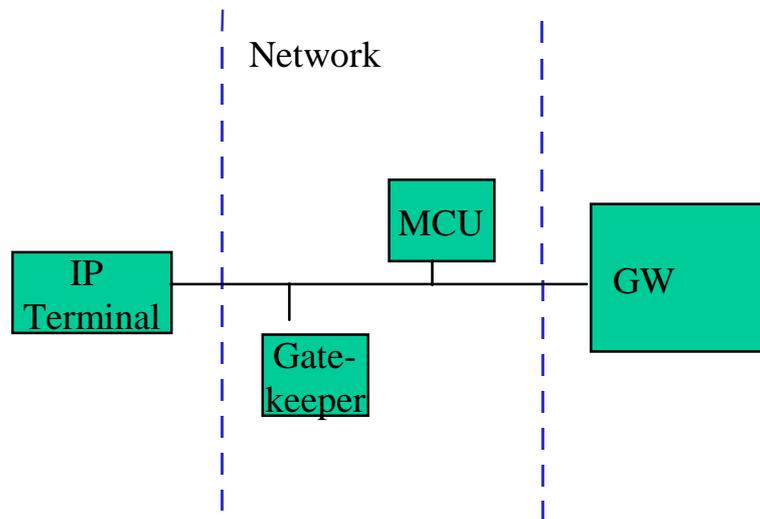


Figure 2-14: H.323 entities.

2.6.6.1. Call models and call setup in H.323

H.323 call signaling can be either directly between endpoints (terminals) or it can be gatekeeper routed, in which case the gatekeeper relays all signaling between endpoints. The routed signaling is needed for terminals that do not contain an multipoint controller (MC), that provides conference control (e.g. establishment of common communication mode and media channels).

The call setup takes place in several steps (see Figure 2-15: Simple call setup signaling.). If the implementation of H.323 uses gatekeepers, the calling endpoint must first request permission to place the call from the gatekeeper using H.225.0 ARQ (1). If the call is allowed the gatekeeper responds with ACF (2). A rejected call response is ARJ (2). If no gatekeeper is used, or after permission the endpoint 1 sends a Setup (3) to endpoint 2. The called endpoint acknowledges the call setup with Call Proceeding (4). If the called endpoint is able to accept the incoming call and the implementation uses a gatekeeper, it must request permission to receive a call with the same ARQ (5) /ACF/ARJ(6) procedure as above. If the gatekeeper allows the call to be received, the called endpoint sends an Alerting (7) message to the calling endpoint indicating that the user is notified of the incoming call. If the user answers the call, the called endpoint sends a connect message to the calling endpoint[9].

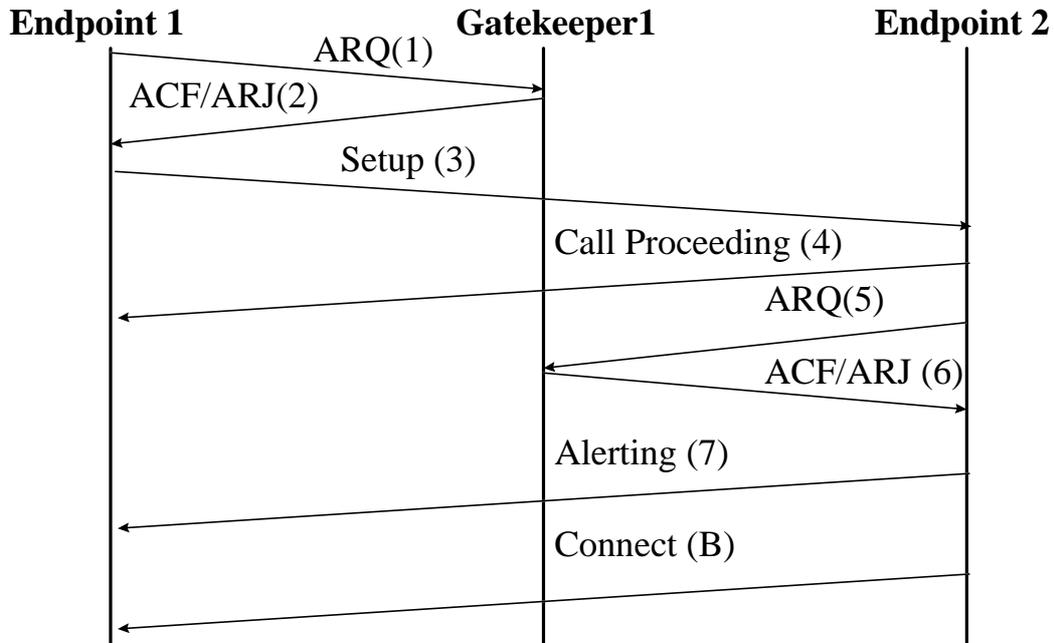


Figure 2-15: Simple call setup signaling.

The communication between H.323 terminals takes place over logical channels after call setup. The logical channels are opened using procedures defined in H.245. H.245 control takes always place on logical channel 0. Separate logical channels are opened for audio, video and data. Multiple channels can exist for each media type through the use of H.225.0. Data communications typically use T.120 series recommendations to define procedures, protocols and applications.

2.6.6.2. H.255.0

H.225.0 covers protocols and message formats. It is designed to operate over various LANs, such as IEEE 802.3 and IEEE.802.5. Acting as a convergence layer, above the transport layer, H.225.0 is protocol independent and can be used over LANs with QoS guarantees as well. The scope of H.225.0 is: *communication between H.323 terminals and gateways in the same LAN using the same transport protocol*. H.225.0 may be used over interconnected LANs or even over the Internet, but the performance is acceptable only when the network load is low. The H.323/H.225.0 protocol stack is depicted in Figure 2-16.

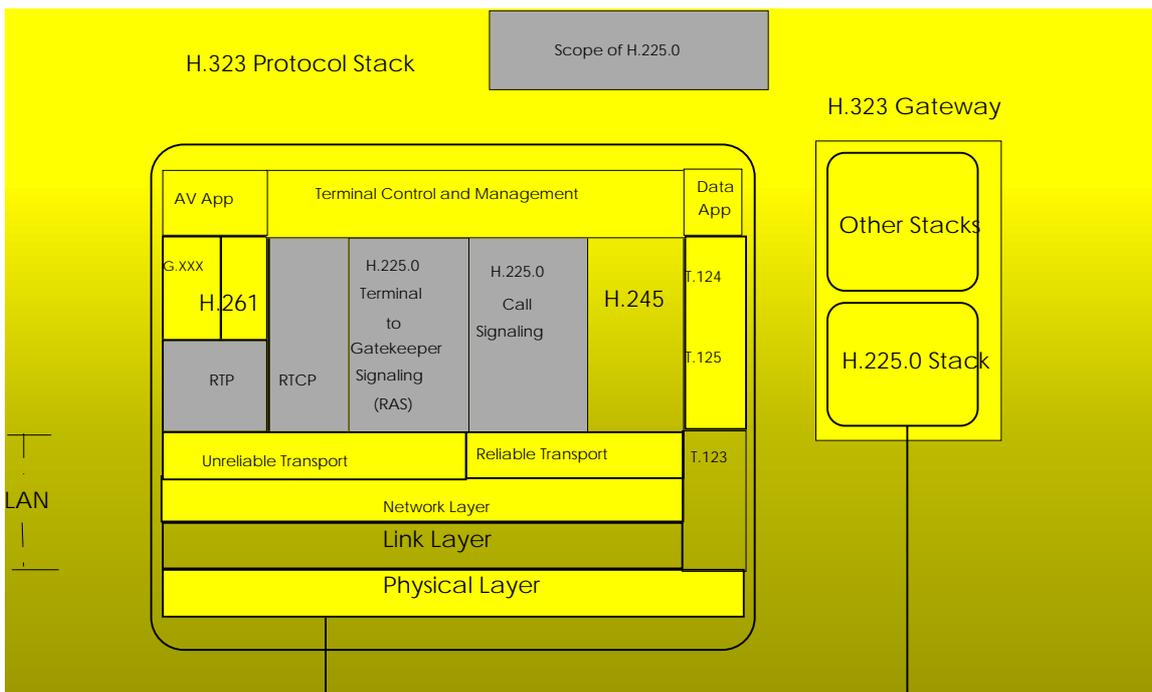


Figure 2-16: The scope of H.225.0.

In H.323 audio and video packets are formatted as defined in H.225.0 using IETF RTP. RTP payload format specifications are used for specifying the carriage of audio and video in RTP. A separate logical channel for RTCP QoS feedback to the source of media is also opened. The feedback information can be used by the media source to adapt encoding or buffering schemes.

2.6.7. IP Switching

The rapid growth of the Internet has resulted in congestion in many areas of the Internet. At the same time access speeds to Internet are increasing and thus new IP routing capacity is needed. On the corporate side, a typical user has a 10BaseT Ethernet LAN connection. LANs are increasingly divided into workgroup sized sub-nets forming a campus network, where routing is used to connect individual sub-nets and the sub-nets to the corporate intranet and to the Internet. This requires some heavy-duty routing at wire speeds. A number of incentives to deliver high-speed routing are underway: the Multi-gigabit routing, IP/ATM [10], the Cell Switch Router (CSR)[11], Tag Switching [12], Aggregate Route-based IP-switching [13], Multiprotocol Label Switching (MPLS) [14] and Multiprotocol over ATM (MPOA)[15]. Here we will present one of these Ipsilon's IP switching[16], [17], the only one available at the time of writing as a commercial product.

2.6.7.1. The Architecture of IP Switching

The components of a router are: line-cards to physically interface the backplane bus to the links, switch fabric to interconnect the various components of the router, forwarding engine and a network processor. The network processor runs the routing protocols and computes the routing tables, and it typically processes the packets needing special handling. The forwarding engine inspects the packet headers and decides which line-card the packet should be forwarded to. In high-speed routers the backplane is replaced by a switching fabric, which offers a much higher aggregate capacity.

IP switching differs from the routing approach in that IP switching makes the routing decision per flow based not packet per packet. Once a flow, a sequence of packets with common properties is identified, it is switched through the switch fabric, and all the subsequent packets of the flow traverse the switch fabric without having to go through the forwarding engine. The switch fabric used with IP switching is an Asynchronous Transfer Mode, ATM switch. The same approach of using ATM architecture to speed IP routing is used in [10], [11], [12],[13],[14] and [15].

One of the functions of flow classification is to select the flows to be switched through the ATM switch and those that should be forwarded through the forwarding engine. The flow classification decisions can be based on e.g. source and receiver IP addresses and TDP and UDP port numbers. Different IP flows may require different QoS. Ipsilon intends to add QoS guarantee capabilities to its switches later. The QoS reservations would be signaled from the application to the IP switches using IETF's RSVP, see 3.4.1.

In the ATM switch the classified flows that have been decided to be switched are assigned to a separate VC. ATM switching requires that all traffic must be labeled with a VCI indicating the VC it belongs to. The information of the association between a flow and VCI label is distributed upstream using Ipsilon Flow Management Protocol, IFMP. Another protocol, the General Switch Management Protocol is used for switch control between the ATM switch and the IP switch controller implemented as separate units. This is depicted in Figure 2-17.

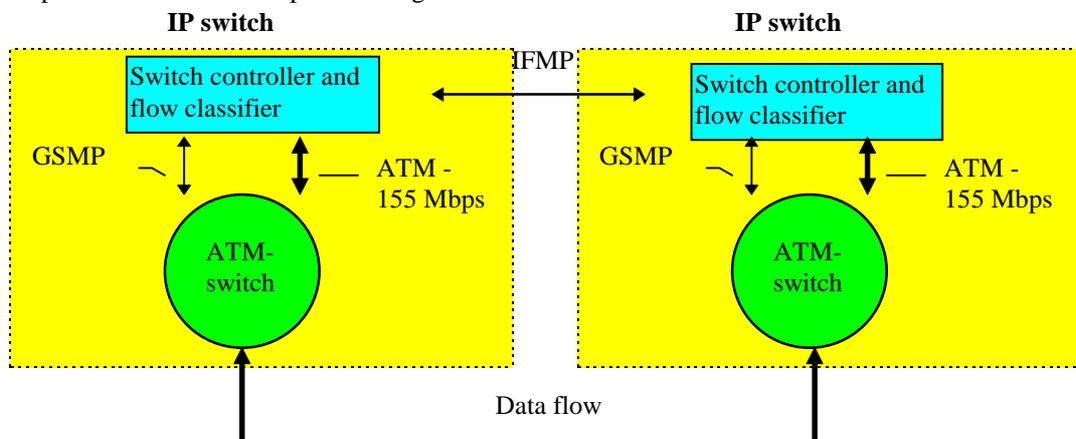


Figure 2-17: The structure of IP-switches and the protocols.

The protocol hierarchy of IP switching is in Figure 3-1. The Ipsilon solution uses AAL-5. LLC/SNAP encapsulation for IP packets is only used on the default channel. On the flows redirected to a specific VC the LLC/SNAP and IP header fields are removed. The ATM ARP and LES - protocols have been replaced with Ipsilon's own IFMP.

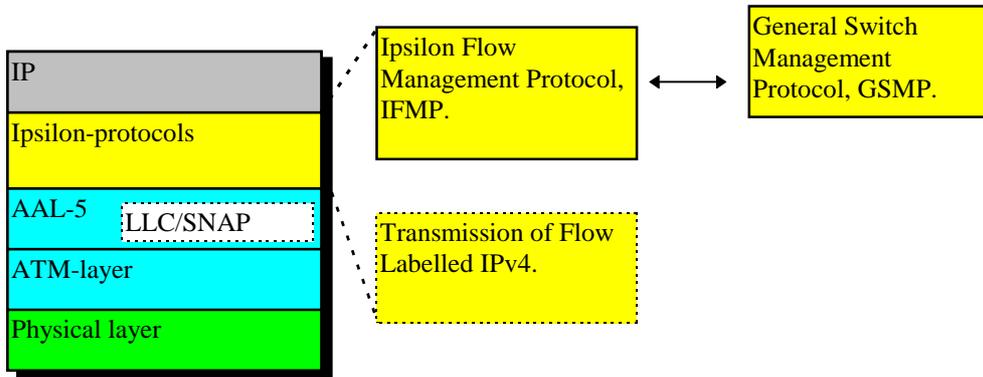


Figure 2-18: The Ipsilon protocol hierarchy.

2.6.7.2. IP Switch Operation

In this section we will briefly go through the operation of IP switching. A very good presentation of the operation and performance issues of IP switching can be found in [18]. The senders and receivers considered here can be either other IP-switches or terminals. The sender is located upstream from the switch and the receiver downstream. When a new IP packet X enters the ATM switch fabric of the IP switch it uses the default connection (VP=0, VC=15). From the ATM switch the packet is directed to the connection X' to the control processor. Using pre computed routes and routing protocols the packet is forwarded to the appropriate default connection W on the upward link, See Figure 2-19

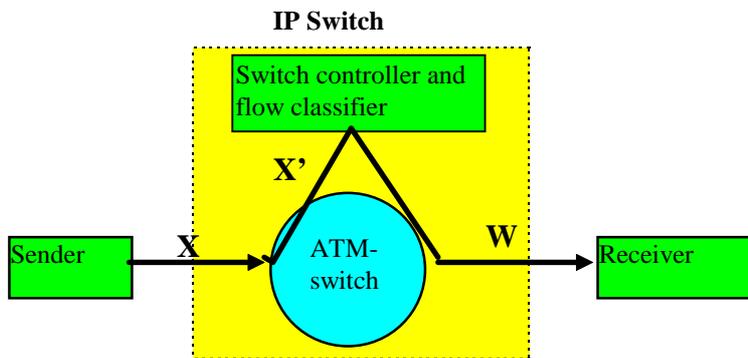


Figure 2-19: The IP switch as a packet forwarding router.

If in the process of packet forwarding the flow classifier identifies a flow (see Figure 2-20), the flow is switched to a separate virtual channel. First the switch controller using GSMP opens a new connection between Y and X', then using IFMP it signals upstream to the closest neighboring element a request to direct the flow to the VC Y. The redirect request is not acknowledged. The first packet on the new connection is an indication of successful redirect. From this on all subsequent packets flow through Y to X', and the routing information is cached thus speeding the forwarding of packets.

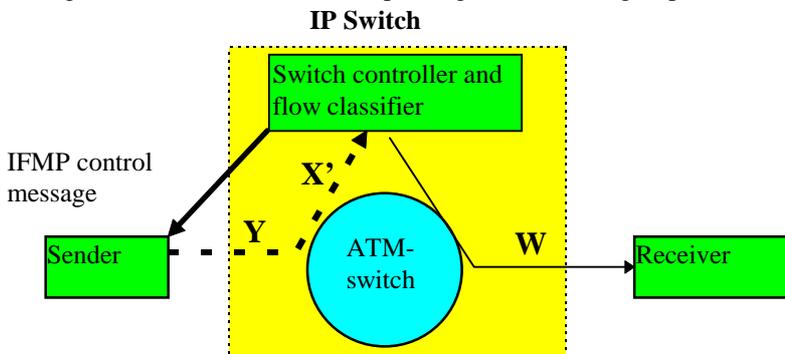


Figure 2-20: Soft-state routing and flow control.

We further consider the situation of Figure 2-20. Now, if the downstream closest neighboring element (the receiver) sends an IFMP requests to direct the flow on connection W to a new connection Z (Figure 2-21), and the flow is redirected on that connection, we have a direct connection through the ATM switch.

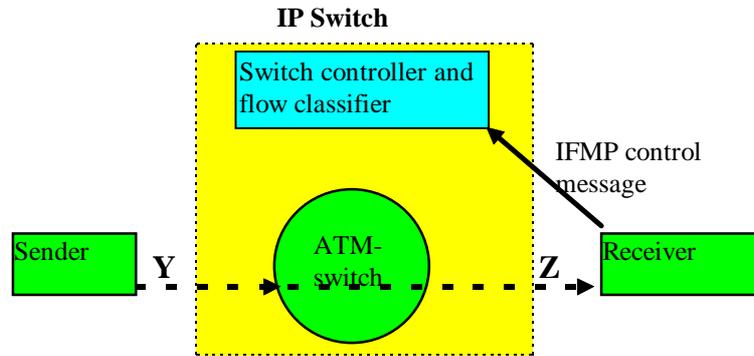


Figure 2-21: A switched flow in IP switching.

An IP switching network is in Figure 2-22. If we now consider a case where the IFMP connection requests have been issued first from the neighboring element of switch₁ to switch₁, then from switch₁ to switch₂ and from switch₂ to the edge device we have the situation of Figure 2-21 in both of the switches for some particular flow. The edge device will insert a corresponding VCI in all packets of the flow, and the headers of the packet are formatted according to some flow type. The flowtypes are in Table 2-2.

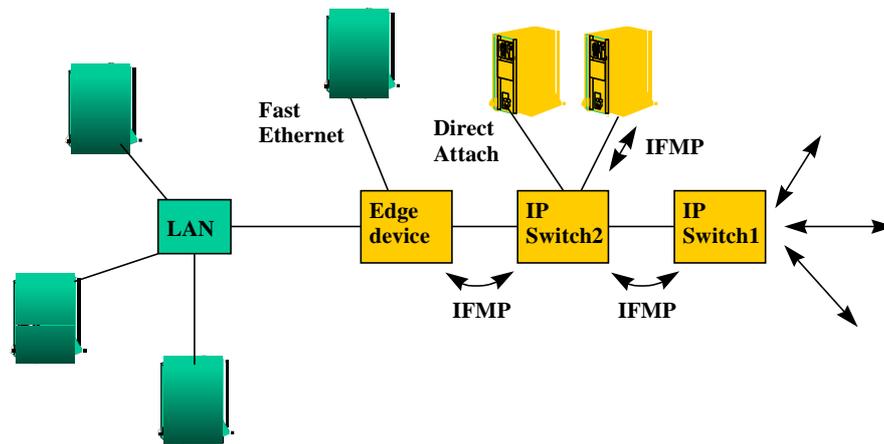


Figure 2-22: IP switching network.

2.6.7.3. QoS of IP switching

The QoS of IP switching is policy based as opposed to RSVP and ATM contract-based QoS, where the user requests a certain degree of QoS. The requested QoS of the connection is defined e.g. in terms of end-to-end delay, average throughput, delay variation etc. The policy basis means that the network manager sets policies according to which the traffic is given QoS differing from that of the default connection. The policies can be based on IP addresses, IP ToS, TCP/UDP port, or a combination of the three. Different QoS is achieved with:

1) Flow actions

The flow can be cut through the switch or it can go through the default path through the processor.

2) QoS actions

A traffic flow can receive four different priorities: high, medium, normal and low. In addition to this rate shaping can be used.

Table 2-2: Flow types.

Flowtype	Contents	Adaptation	MTU	Control
Default	TCP/IP	LLC/SNAP + AAL-5 CPCS-PDU	var	Default
Type 0	TCP/IP	AAL-5 CPCS-PDU	1500 bytes	IFMP Flow Type 0-redirect message
Type 1	modified TCP/IP	AAL-5 CPCS-PDU	1484 bytes	IFMP Flow Type 1-redirect message

Type 2	modified TCP/IP	AAL-5 CPCS-PDU	1492 bytes	IFMP Flow Type 2-redirect message
--------	-----------------	----------------	------------	-----------------------------------

IP switching uses Class-Based Queuing (CBQ), a combination of priority scheduling and FIFO queuing with a weighted scheduler. CBQ uses FIFO queuing within a class. When a source (a flow) exceeds its target rate the excess traffic is discarded or put to a buffer called waiting room, i.e. incoming flows go first through a class policer. The packets conforming to the rate and the packets in the waiting room are then run through a WRR scheduler to the priority scheduler, from where they are forwarded to the link.

The IP switching way of improving QoS has been argued to have draw-backs, even to the extent that no improvement in QoS might be seen. At least four types of problems could occur:

1. The switch controller might select a route that can not meet QoS requirements. This could happen because OSPF and RIP that do not support QoS routing are used;
2. Traditional packet forwarding is used until all the switch controllers along the route have established a virtual circuit path. The performance before this can be poor;
3. The current implementation of IP switching only supports the limited IP header ToS for defining QoS. Applications in general can not take use of this;
4. No mechanisms for ensuring that all cut-through stubs in each switches along the path are offering the same level of QoS, i.e. there are no guarantees of a consistent level of end-to-end quality

2.7. Speech coding

Speech coding is conversion of a speech signal to a digital representation. The simplest way to do this is by applying sampling theorem directly. This means sampling the waveform at a rate of twice the highest frequency present in the signal, and then digitizing the resulting samples to some desired degree of accuracy. The telephony nominal bandwidth is 4 kHz, so the speech signals need to be sampled at a rate of 8000 samples per second. The desired signal-to-noise ratio is dependent of the encoding precision, usually either 8 or 16 bits per sample. The total rate is thus either 64000 or 128000 bits/s [19].

One of the goals of speech coding is to reduce the bit rate. To do this speech waveform specific properties have to be exploited. Adaptive quantizers vary their characteristics over time. This is to match the dynamic range variations of the speech signal. Time-varying filters exploit the short-term and long-term correlations of the signal. Coding methods can also take advantage of the property that in human hearing noise can be masked by the speech signal, if the spectral level of the noise is below the spectral level of the speech.

A typical speech coder consists of two modules: an analysis module and a synthesis module. The Analysis module extracts from the speech waveform the time varying excitation waveform and the time varying filter parameters. The Synthesizer module recreates the perceptually best match to the original speech waveform. The most common speech synthesis models are the LPC *vocoder* model, the *multipulse* model and the *stochastic* model. The vocoder or voice coder is the most traditional of these models. Examples of the multipulse and stochastic coders are the MPLPC, multipulse linear predictive coder, and CELP, the code excited linear prediction [19].

2.7.1. Speech and Audio Coders for VoIP and their Quality

The design parameters of all speech coders are bit rate, quality, signal delay and complexity. The bit rate is a measure on to what extent the speech model has been exploited in the coder. The lower the bit rate the greater the reliance on the speech production model. Quality is a measure of degradation of the speech signal and is measured in subjective tests of speech intelligibility and naturalness [19]. Signal delay is a measure of the duration of the speech signal used to estimate coder parameters in both encoding and decoding, plus the delay of the transmission channel. The longer the allowed delay in the coder, the better it can estimate the synthesis parameters. Complexity is a measure of the computation required to implement the coder algorithms in hardware or software. An ideal coder would have a low bit rate, high perceived quality, low signal delay and low complexity. Real coders make tradeoffs among the attributes.

The quality of speech coders is measured in subjective listening tests. The diagnostic Rhyme Test (DRT) measures intelligibility of speech in terms of distinguishing minimally distinct pairs of rhyming words. Intelligibility for coder bit rates from 64 kbit/s to 4.8 kbit/s is close to that of natural speech. At lower bit rates a slight degradation is observed.

The MOS test of speech quality uses a five point scale:

- 5, excellent quality, no noticeable impairments
- 4, good quality, only very slight impairments
- 3, fair quality noticeable but acceptable impairments
- 2, poor quality, strong impairments
- 1, bad quality, highly degraded speech

The MOS scores for a coder are derived by averaging the responses of a large number of listeners. The scores are highly variable from test to test. Usually a high quality speech signal is used as a reference to reduce the variability. High quality reference signals have MOS scores from 4.0 to 4.5. There are significant differences in MOS scores for different telephone bandwidth coders. Coders with bit rates from 64 to 16 kbit/s have all relatively high MOS scores, 4.0 or higher. At around 8 kbit/s the scores fall below 4 to around 3.5 to 3.8 and at 4.8 to 2.0 kbit/s the scores fall to 2.0 to 3.0. For network applications the score of 4.0 is considered the low limit. Scores below 3.0 are considered acceptable only to military applications.

For VoIP applications one important design issue of the coder is its the robustness to packet loss.

Schemes that extrapolate some coder parameters when a speech frame is lost have been developed. The schemes only take effect in the case of a lost voice packet [20]. Such modifications do not increase the complexity of the coder, but robustness to frame erasures is substantially improved.

Table 2-3:Speech band coder comparison.

Coder standard	Algorithm	Complexity % 100MHZ Pentium / MIPS (DSP) [19], [21]	Frame size/ lookahead (ms) [21]	Codec delay using DSP (ms) *	Comp.	Rate (kb/s) [21]	MOS (Clear channel, single encoding) [22] **
PCM	PCM	0 MIPS		0	1	128	4.5
G.711	PCM μ A-law	0 MIPS	0.125 /0	0.225	2	64	4.10
G.726 (G.721)	ADPCM	<1% / 1 MIPS	0.125/0	0.225	8/5.3/4 /3.2	16/24/32/4 0	3.85
G.728	LD-CELP	65% / 30 MIPS	0.625/0	3	8	16	3.61
G.729 (IS-641, GSM EFR)	CS-ACELP	50% /20 MIPS	10 /5	25	16	8	3.92
G.729A	CS-ACELP	25-30% / 11 MIPS	10/5	25	16	8	3.7
G.723.1	ACELP&M Q-CLP	35-40% /16 MIPS	30/7.5	67.5	24.2/2 0.3	5.3 / 6.3	3.9
GSM 06.10	RPE-LTP	10 MIPS	20/0	40	9.7	13	3.50
IS-54 (TIA)	VSELP	24 MIPS	20/5	45	16	8	3.54
IS-96 (TIA)	QCELP		20/5	45	15/32/ 64/160	8.5/4/2/0.8	
	LC-CELP	10 MIPS			8	16	
	CELP+	30 MIPS			18.8	6.8	
	RCELP	16 MIPS			26.7	4.8	
FS-1016	CELP	30 MIPS			26.7	4.8	3
	TFI	150 MIPS			32/53. 3	4 /2.4	
FS-1015	LPC10E	15 MIPS			53.3	2.4	2.4

*) Note that this only includes the delay of the algorithm, i.e. lookahead and the frame delay in both ends. The total system delay would also include processing delay for computation, multiplexing delay, buffering delay and transmission delay. For example a GSM cellular system using GSM 06.10 coder has a system delay of 76.7 ms.

**) The mean opinion scores for coders under background noise and over degraded channels vary.

Particularly the low bitrate coders (<16 kb/s) have been in the past sensitive to background noise. The newer cellular coders, such as G.729 (IS-641) give better scores than the old (e.g. GSM06.10) in these conditions.

2.8. Voice Source Characterization

IP-networks integrate a wealth of different services and traffic types: packetized voice, packet video and data traffic. Traffic theorists have come up with sophisticated traffic models to carry out accurate design and performance evaluation of the packet and cell switched networks. Bursty sources have been used as models for data traffic; file transfers and image transmission. Data traffic in general is

characterized by alternating, randomly varying periods of inactivity and activity, and the periods of activity are much shorter than periods of inactivity. Measurement studies in the Internet have shown that the aggregate traffic of these different sources is self-similar or fractal-like in nature[32]. With self-similarity is meant that the traffic is invariant of the time scale: if we look at the traffic in the scale of seconds and hours the statistical properties do not change. The aggregate traffic models of different types of traffic sources, and the performance evaluation using the models is a complex issue. We will not go very deep in to the challenging, and highly interesting world of queuing theory, but instead will present the voice source models at a very basic level.

The standard coding methods of a voice source result in more or less what is depicted in Figure 2-23. The voice source is coded so that the traffic is continuous bit- and packet-rate. The packet length is dependent of the codec used, sampling frequency, frame length and the number of coded frames in one packet. Frame length indicates the number of samples per coded frame. For example if we are sampling 8000 samples/s and the frame length is 20 ms we have 160 samples in a frame. The packetization interval is constant and typically 1-2 frame lengths, i.e. 20-40 ms.

A single voice source can be represented by a two-state process. Human speech consists of alternating intervals of inactivity (silence) and activity (talkspurt). The talkspurt lengths average from 0.4 to 1.2 s, and the silence average 0.6 to 1.8 s. This phenomena has been used for a long time in analog telephony to multiplex and pack multiple calls into one trunk in systems called time-assigned speech interpolators, TASIs, and the digital telephony counterparts, DSIs.

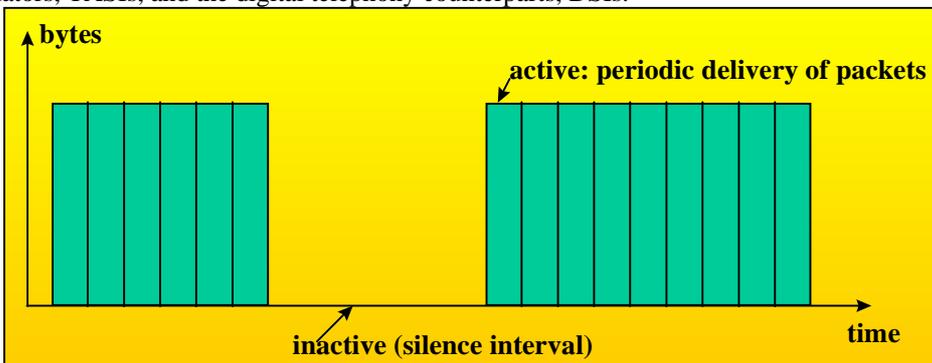


Figure 2-23: The traffic characteristics of a voice stream.

The speech source states can be approximated to be exponentially distributed in length. The parameter λ presents the transition from the inactive state to the active state, and the parameter α presents the rate of transition from the active state to the inactive state. This two-state birth-death model is depicted in Figure 2-24. The average talkspurt length is $1/\alpha$ seconds, and similarly the average silence length is $1/\lambda$ s. The probability that a speaker is active, or the speaker activity factor is $\lambda/(\alpha+\lambda)$ [32].

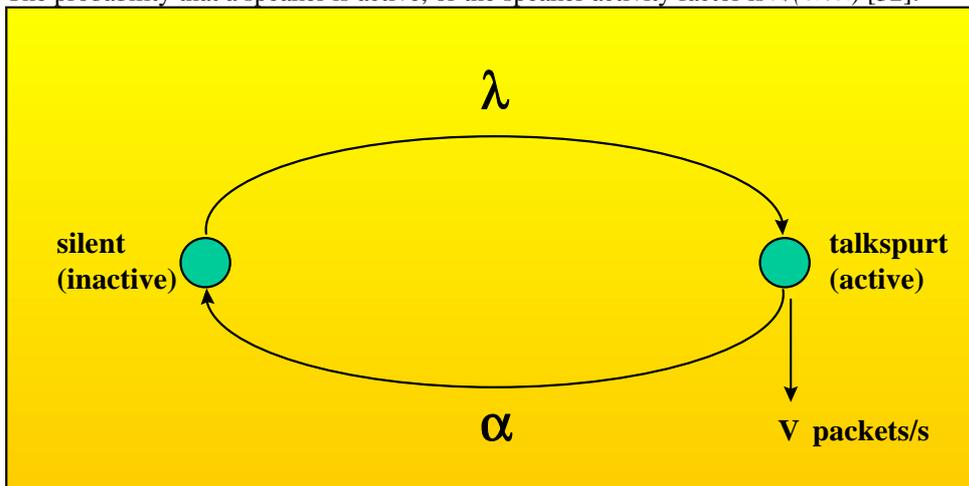


Figure 2-24: Two-state model of a voice source.

2.8.1. Packetization Process

Implementations of VoIP terminals may vary, but they all do the following:

1. Audio from microphone or line input is A/D converted at audio input device, with a sampling frequency of 8 kHz / 16/8 bits;
2. The samples are copied to a memory buffer in blocks of frame length;
3. The VoIP application estimates the energy levels of the block of samples
4. Silence detector decides whether the block is to be treated as silence or as part of a talkspurt;

Voice over IP Concepts

5. If the block is a talkspurt it is coded with the selected algorithm (e.g. GSM 06.10)
6. If this is the beginning of a talkspurt some bits are added;
7. A chosen number of blocks of audio are added to create one RTP packet, and RTP headers are added;
8. Packet is written to correct socket interface (UDP port), IP-headers are added, physical framing and transmission;
9. Packet is received, de-framed, IP-header checked;
10. Packet is read through the UDP socket;
11. RTP-headers are checked for type of payload data, sequence number, timestamp;
12. Sequence number and timestamp are used to detect reordering and duplicates;
13. The insertion point of the incoming audio data is determined in the playout buffer;
14. A block of audio is decoded into samples using the same algorithm it was coded; with and inserted in the playout buffer;
15. The block of samples is copied from the buffer to the audio output device;
16. The audio output device D/A converts the samples and outputs them.

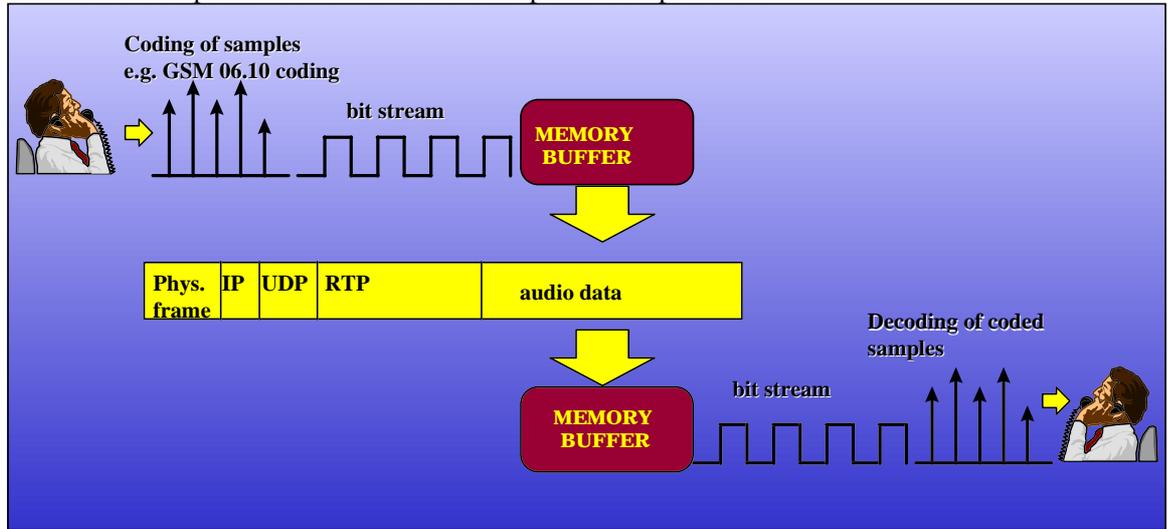


Figure 2-25: From voice to bit stream and back.

The packetization and de-packetization is depicted at a very high level in Figure 2-25. In addition to the functions mentioned above the terminal sends congestion information and receives feedback information. Also an automatic gain and echo cancellation can be implemented in the audio input. The buffering can take place both after sampling and after coding at the sender. The receiver can buffer both after receiving and before D/A-coding.

3. VOIP QUALITY OF SERVICE ISSUES

This chapter handles the mechanisms relating to the quality of voice over IP. The quality issues of VoIP are complex and challenging. To start with, today's networks provide only limited QoS capabilities at LAN level, and the Internet is fundamentally a shared architecture with very limited implemented QoS capabilities. Still, the network is only a part of the VoIP architecture. We should focus more on the end-to-end quality of VoIP, which is the factor of the quality of all the pieces of the communications chain. The end users of VoIP will and should not be bothered by the technical terms, the technologies and the parameters underlying and defining the QoS of a connection. For the end user the perceived quality is the degree of satisfaction, not a set of parameters.

3.1. What Is QoS

The CCITT Recommendation E.800 gives a generic definition of QoS:

“The collective effect of service performance which determine the degree of satisfaction of a user of the service[23]”

ETSI recommendation ETR300 003 further sliced and refined the ITU definition in to sub-definitions, that correspond to the requirements and viewpoints of the different parties taking part in the communication:

- QoS requirements of the user/customer
- QoS offered by the service provider
- QoS achieved by the service provider
- QoS perceived by the user/customer
- QoS requirements of the Internet Service Provider

If we deeper into the QoS at a more technical level, the quality issues of VoIP fall under three concepts [24]:

1. The characteristics of the Quality of Voice over IP
The characteristics of QoS are quantities that express the impairments caused by the terminals and by the network. Examples of these quantities are processing delay, transfer delay, delay variance and packet loss.
2. QoS management
QoS management defines the control mechanisms in a packet network and the mechanisms the applications use to adapt to the changing conditions of the network.
3. QoS agreements
QoS agreements are step away from the best-effort service. Current efforts in R&D try to extend present protocols and switching disciplines to support desired requirements on the quality of service.

3.2. QoS Characteristics

QoS Characteristics are quantities expressed by:

- throughput of a transfer when the IP-packet is transferred from source to receiver
- processing delay. The processing delay includes the time required to detect, process and code the analog voice signal; compress and packetize it. Ideally the formation of an IP packet should not take longer than the time to send a PCM on a PSTN from a telephone set;
- transfer delay of IP packet between the sender and the receiver. If there are intermediate nodes the delay will be the sum of all nodal transfer, buffering and processing delays;
- error rate of a transfer;
- packet loss;
- transfer delay variance;
- nature of the communication (half/full duplex).

3.2.1. End-to-end Transfer Delay

The network delay is the difference between the time when a packet is received at the receiver and the time of transmit at the sender. End-to-end delay of VoIP should also take into account the coding and buffering delays of the terminal equipment.

Network delay is introduced in several points. There is a large fixed delay caused by the propagation of the packet at the speed of light and delay caused at each switch point. Before retransmission at each intermediate node the whole packet needs to be received. Additional delay is caused by different service

queues in the network switches. This delay is variable and must be bounded and minimized for adequate realtime service [25].

The components of end-to-end delay of VoIP are [26], [27]:

1. Framing delay
2. Coder delay
3. Processing delay at the terminal
4. Buffering delay at the terminal
5. Packetization delay T_p (at terminals and gateways)
6. Nodal processing delay
7. Nodal queuing delay
8. Propagation delay

Framing delay represents the amount of speech frames included in a voice packet. The framing delay is a multiple of the coder block length, which represents the number of samples coded at one time. For example the block length can be 20 ms and the framing delay 40ms i.e. 2 blocks of samples per packet. Coder delay is caused by the frame size, lookahead and coder algorithm processing. When this delay is added on top of already large network caused delay, it can be significant. The coding delay is essentially a design parameter of the codec, and as such out of the scope of this thesis.

The processing delay in the terminal is caused by the detection, processing and coding of the analog voice signal, compressing it and building an IP packet. The processing delay is dependent of the speech codec in use. Some codecs like the G.711 PCM and ADPCM have small delay, while coders that require more processing (e.g. GSM06.10 and G.723.1) introduce more delay.

Buffering delay at the terminal is caused both at the sender and at the receiver. The sender buffers segments of samples before transmission in order not to lose any, and the receiver buffers frames in order to remove network caused interarrival jitter of the packets.

The purpose of the following presentation is to define the total end-to-end delay for a packet traversing k links sufficient to allow 95% of packets to be delivered with this delay. The presentation is from [26].

Presentations of related problems can be found for example in [28, 29, 30, 31, 32].

Packetization delay can be expressed:

$$T_p = I_p/R \quad (1)$$

, where I_p = information bits / packet, R = voice data rate in bits/s.

Transmission efficiency is the ratio of bits going through the link divided by the capacity of the link. If the capacity of the link is ρ , λ is the offered traffic in packets/s and μ is the link capacity in packets/s then we have:

$$\rho = \lambda/\mu \quad (2)$$

The nodal processing delay is a function of processing capability built into the switches, and in general does not contribute materially to end-to-end delay. Hence it is omitted in presentation. The third factor, nodal queuing delay represents the time spent by a packet in an outgoing queue at a switching node waiting to be transmitted. The statistics of packet arrival at any given node is a complex function of the delays encountered at previous nodes and the voice packetization process, since the voice traffic is typically an on/off type with periods of silence t_p , when no packets are generated, and periods of talkspurts. The process can be approximated with an exponential distribution. Thus the queuing delay density function $p(d)$ for a single node based on the above assumptions can be expressed as:

$$p(d) = \mu(1 - \rho)e^{-\mu(1 - \rho)d}, d \geq 0. \quad (3)$$

The performance measure we are going after is the 95th percentile of the cumulative distribution of delay. The packet traverses k -links, so we define D_k as the cumulated delay. Therefore for a single link we have:

$$p(d \leq D_1) = 0.95 \quad (4)$$

D_k is a direct function of $\mu(1 - \rho)$. The delay density equation for one link can be extended to k -links through convolution. We can present the values of D_k for any order of k as:

$$D_k = C_k/\mu(1 - \rho) \quad (5)$$

If we were to assume that there is only voice traffic in the network we can develop this further to get the network delay. A single voice source is capable of transmitting $1/T_p$ packets/s. If then assume that there are an average of N active voice sources on the network, and that the speech is only active half the time (silence suppression used), we have:

$$\lambda = N/2T_p \quad (6)$$

Combining the equations (4) and (2) with (5) we get:

$$D_k = \frac{2C_k T_p}{N} \left(\frac{\rho}{1-\rho} \right) \quad (7)$$

And since the total network end-to-end delay is the sum the packetization delays and the queuing delay:

$$Network_delay = T_p \left(\frac{2C_k}{N} \frac{\rho}{1-\rho} + 1 \right) \quad (8)$$

The total end-to-end delay would take in account all delays incurred between the sender's microphone and the receiver's speaker

3.2.1.1. Packet Length Versus Delay

RFC0889: Internet Delay Experiments handles the relationship between packet length and delay. The experiment with ICMP-packets shows a strong correlation between delay and length, with the longest packets showing delays two or three times the shortest. On paths via ARPANET clones the delay characteristic showed a stronger correlation with length for single packet messages than for multi-packet messages, which is said to be consistent with the design which favors low delays for short messages and high throughputs for longer ones [33].

3.2.1.2. Delay Variance

Variable speech burst delay modulates the duration of silence intervals between speech bursts. The varying of delay or jitter in literature is prevalent in asynchronous networks. Delay variance is defined to be the *maximum difference between end-to-end delays experienced by any two consecutive packets*. Delay variance can introduce gaps in the continuous playback of data or it can shorten the playback of some logical data unit [34].

3.2.2. Throughput

The throughput of a network is the fraction of nominal network bandwidth that is actually used for carrying traffic. In the calculation of this value packet headers are considered as useful data[35]. The throughput a single user gets is some fraction of the network throughput in a shared architecture. What the fraction actually is depends on the type of network, fairness of the connections sharing the network etc. The required bandwidth of VoIP depends on the used coding, the number of frames per packet, and things like is RTP header compression used. In Table 3-1: Required bandwidths if no header compression is used. is a summary of the required net bandwidths using different codecs with one, two and four 20 ms speech blocks per packet, and no header compression.

Table 3-1: Required bandwidths if no header compression is used.

	IP/UDP/RTP (bits)	bitrate bits/s	Sampling rate (Hz)	Packet length (s)	Required bandwidth (bits/s)	No in 64k
PCM	320	64000	8000	0,02	80000	0,8
ADPCM	320	32000	8000	0,02	48000	1,3
LD-CELP	320	16000	8000	0,02	32000	2,0
CELP	320	8000	8000	0,02	24000	2,7
CELP	320	4800	8000	0,02	20800	3,1
LPC10E	320	2400	8000	0,02	18400	3,5
RPE-LTP	320	13200	8000	0,02	29200	2,2
RPE-LTP	320	13000	8000	0,02	29000	2,2
VSELP	320	7950	8000	0,02	23950	2,7
CELP+	320	6800	8000	0,02	22800	2,8
PCM	320	64000	8000	0,04	72000	0,9
ADPCM	320	32000	8000	0,04	40000	1,6
LD-CELP	320	16000	8000	0,04	24000	2,7
CELP	320	8000	8000	0,04	16000	4,0
CELP	320	4800	8000	0,04	12800	5,0
LPC10E	320	2400	8000	0,04	10400	6,2
RPE-LTP	320	13200	8000	0,04	21200	3,0
VSELP	320	7950	8000	0,04	15950	4,0
CELP+	320	6800	8000	0,04	14800	4,3

PCM	320	64000	8000	0,08	68000	0,9
ADPCM	320	32000	8000	0,08	36000	1,8
LD-CELP	320	16000	8000	0,08	20000	3,2
CELP	320	8000	8000	0,08	12000	5,3
CELP	320	4800	8000	0,08	8800	7,3
LPC10E	320	2400	8000	0,08	6400	10,0
RPE-LTP	320	13200	8000	0,08	17200	3,7
VSELP	320	7950	8000	0,08	11950	5,4
CELP+	320	6800	8000	0,08	10800	5,9

3.2.3. Packet Loss

Packet loss is a serious impairment of the Internet of today on realtime communication. Packet loss as high as 10% to 30% is not uncommon. Non-real time traffic uses TCP to resend lost packets, but resending packets causes the playback points of the packets to be missed in realtime communications. For realtime voice over IP a packet is considered lost when its playback point is missed. In literature speech clipping of 16 to 64 ms has been noted to cause noticeable quality degradation unless the percentage of speech clipped is under 2%. For higher percentages of speech clipped at the same clip lengths the quality degrades, but intelligibility is still maintained. For speech clipping durations of >64 ms the quality of the speech is degraded seriously and intelligibility is reduced [36].

The effect of packet losses on the perceived quality of speech is dependent of the packet size. In [37] listening tests suggested that in the sense of robustness to packet losses the optimal packet length is between 16-32ms. The duration strikes a tradeoff between the number of speech losses per second and the probability of totally losing a phoneme.

Packet loss can be characterized with many different quantities. One of the most used ones is the average packet loss, or *unconditional loss probability*. We define l_n the boolean variable set to 1 if a packet is lost, and 0 otherwise. The average loss is thus equal to the expected value of l_n :

$$ulp = E[l_n] \quad (9)$$

The burstiness of the loss process, or the correlation between successive packet losses is not captured with ulp . One way to do it is to consider the conditional probability that a packet is lost given that the previous packet is lost. *Conditional loss probability* is denoted:

$$clp = P[l_{n+1}=1 | l_n=1]. \quad (10)$$

As stated earlier ulp is typically high in the Internet, however for a single audio stream clp remains low [31].

Gaps in the received speech due to dropped or lost packets can be handled with a number of proposed strategies: [38]

1. *Silence substitution*, i.e. replace the discarded packet with samples having zero-amplitude values[37];
2. *Noise interpolation*, i.e. fill the gaps with noise, called comfort noise;
3. *Packet interpolation*, compensation for dropped packets through interpolation;
4. *Repetition of last packet received*; the speech can be classified in different classes, and each class can be sent in a different packet, and the repetition algorithm may vary with the class of the previous packet and with the sequence number;
5. *Pattern matching*, extract packet long segments from previous packets that correspond to the few milliseconds before the detection of packet loss;
6. *Repetition of the last pitch waveform* for voiced segments, otherwise use the previous packet[39].

Of the former pitch waveform replication and pattern matching are novel methods that require more signal processing and memory than packet repetition. Silence substitution and noise interpolation are simple techniques.

3.3. QoS Management

In this section the principles of QoS management for VoIP are explained. The QoS management is currently centered around the use of RTCP for control: delay, packet loss and packet interarrival delay variance measurement and estimation. In best effort networks the main goal of QoS management is providing as good quality as possible and orderly degradation of quality in case of congestion. An example of QoS management is the monitoring of RTCP broadcast messages of a multicast session to detect problems with specific links.

3.3.1. Use of RTCP in Measuring QoS

RTCP sender reports are used for three main purposes: to allow synchronization of multiple RTP streams, to allow the receiver to know the expected data and packet rates and to measure the distance in time to the sender. The most important issue is the synchronization of multiple sources.

The receiver reports are used to measure the QoS of the connection: fraction lost, cumulative packets lost, the extended highest sequence number received and the inter-arrival delay variance. The cumulative number of packets lost and the sequence number are used to compute the packets lost since the last receiver report. This can be used for determining the long term congestion in a LAN. If the state of congestion is higher than the value set by the terminal manufacturer (programmer) then the terminal should reduce the media rate. High interarrival jitter (the interarrival jitter field) and long intervals in sender reports can also be used as indicators of high state of congestion.

3.3.2. Procedures for maintaining QoS

The methods that can be used by voice terminals and gateways to respond to congestion can be grouped in two: those that respond to short term problems and those that respond to longer term problems. The methods do not seek to maintain QoS, but instead to provide an orderly degradation of service. Short term responses are responses to problems like lost or delayed packets. A typical long-term response would be that to the growing congestion on the LAN. The media degradation order is: video, data, audio, control. There are three typical short-term responses: reducing the frame rate for a short period of time, reducing packet rate by mixing audio and video in same packet, and packet rate reduction by video fragmentation at the H.261 macro block level. More sophisticated responses would be increasing the amount of redundancy information in packets or increasing the amount of information used in Forward Error Correction [40, 41, 42].

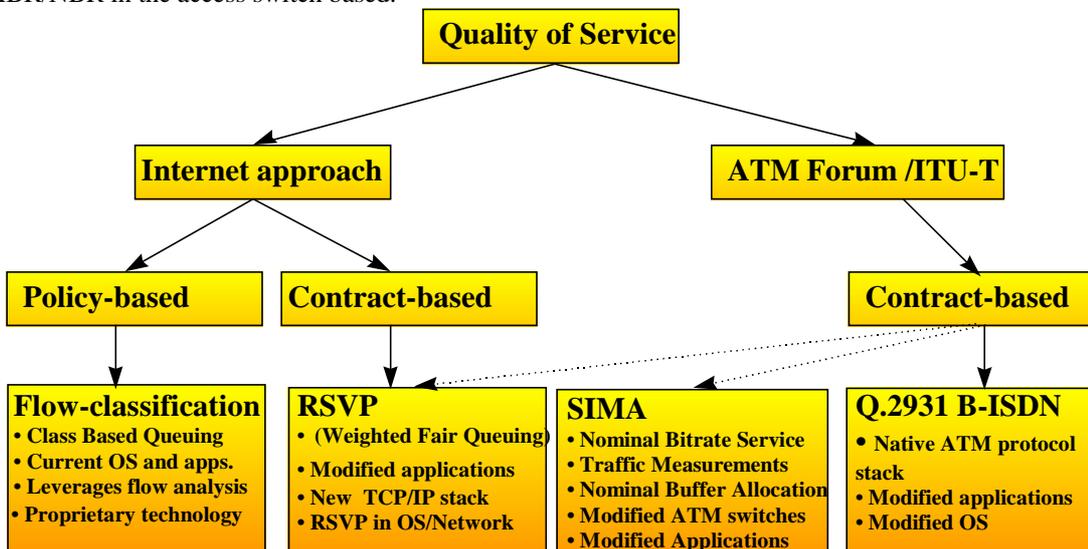
Long-term responses are: reduction of media bit rate, turning of media of lesser importance and returning a busy signal to the receiver as indication of LAN congestion. The busy signal sending can be combined with turning of media.

In a multi-router configuration reacting to delay variance can be difficult. It may be impossible to distinguish the source of delay variance when there is a lot router incurred reordering and varying of the packet delay.

3.4. QoS Agreements

QoS agreements are a step away from the current best-effort model of the Internet. When QoS agreements are made, typically the application request some QoS from the network and the network gives guarantees on QoS. In this case we call the provision of QoS. The ATM Forum traffic classes are an example of this. The IETF has defined a similar contract-based model, the Internet Integrated Services Architecture for the Internet. In the IETF model the guarantees are reserved using RSVP. Another way of giving applications QoS is the policy-based approach (see section 2.6.7).

The Simplified ATM management proposed in [43] and [44], lies somewhere between the ATM Forum approach and the Internet approach. In this model each user is allocated a share of the link, a nominal bit-rate (NBR). The NBR is a service provider-customer contract of a guaranteed bandwidth. The actual bit rate of a connections is measured based on the exponential moving average. The measured bit-rate (MBR) is compared to the NBR, and the cells transmitted are given some priority from 1 to 7 based on the ratio of MBR/NBR in the access switch based.



3.4.1. The Internet Integrated Services Architecture

The Internet Integrated Services Architecture is the IETF's approach for providing Internet Quality of Service. The goal was that realtime services could co-exist with the traditional non-realtime best-effort service in IP networks. The IIS is often mistaken to be the same as RSVP. The big picture includes many other elements, and is more a broad reference model, than just the RSVP.

The core service model of IIS centers around the question of time-of-delivery of packets. The QoS commitments made by the network are related to per-packet delay, bounds on minimum and maximum delays being the sufficient parameters. Applications are grouped in two: realtime applications and elastic applications. Elastic applications always wait for data to arrive, where as realtime applications are time sensitive to data delivery. Realtime applications can be further grouped in two: 1) applications that need perfectly reliable upper bounds for delay 2) and applications that can tolerate and adapt to variations in delay and do not need perfectly reliable upper bounds for delay. The service model for the intolerant applications is called *guaranteed service*. The service model for the tolerant applications is called *predictive service* [45, 46].

The predictive service gives the applications a fairly, but not perfectly reliable bound for delay, which can be calculated with properly conservative predictions of the behavior of other flows. The service also tries to minimize the ex post maximum delay. It does not try to minimize the delay of every packet, but rather it tries to pull in the tail of the delay distribution.

Applications can also adapt to changes in the state of congestion of the network by changing their bit rate and thus their traffic characterization. For example video conferencing application can easily change the coding scheme and reduce the frame rate.

The service model for elastic applications is called best-effort service. Also the terms ASAP (as soon as possible) and datagram service are used. Elastic applications are sensitive to delay - excess delay often shows in poor application performance. However the performance of the applications is more dependent on the average delay than on the delay distribution.

The same scheme that is used for predictive service can provide controlled link sharing. The objective is not to bound delay, but to limit overload shares of the link - thus giving the name for the new service *fair share or controlled load*. The technology behind this service and the other new services is WFQ - Weighted Fair Queuing. WFQ-scheduling in the way it is used in controlled load service is available in commercial routers today, and is used to segregate traffic into classes based on things like protocol type or application.

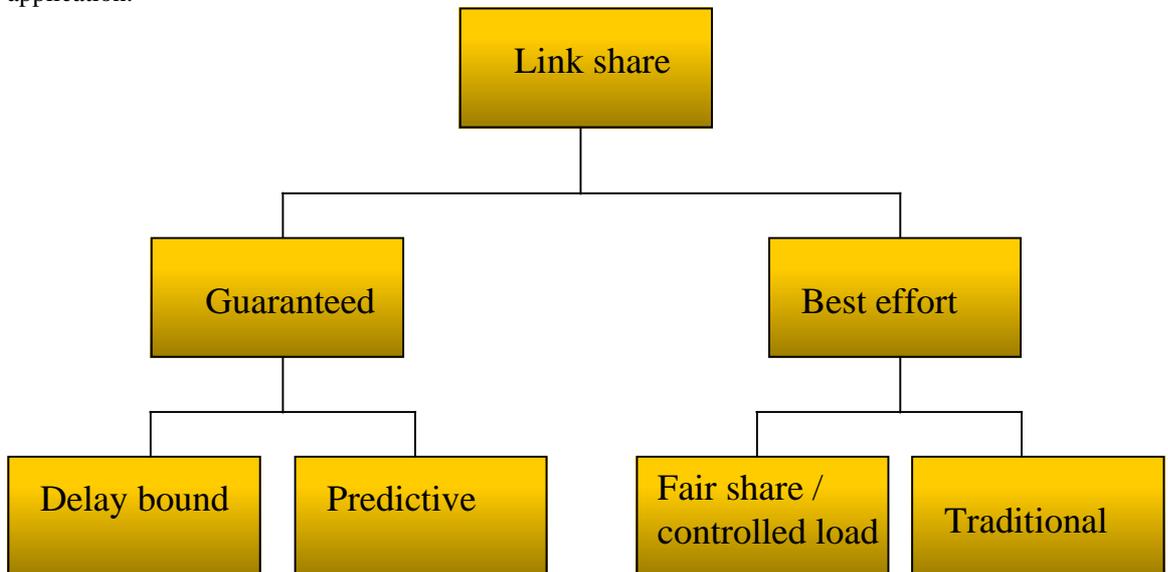


Figure 3-1: IETF traffic service class hierarchy. [47]

4. SYNCHRONIZATION

A packet audio tool periodically gathers a group of audio samples, codes them with a suitable algorithm, packetizes the coded audio and then transmits it to the network. For efficiency the source application transmits data only when there is activity, i.e. audio transmission is divided into “talkspurts” and “silence periods”. The packet network introduces varying amounts of delay on the delivery of the audio packet. This variation is called delay variance. The receiver depacketizes the data and tries to play back the audio as close to the original as it can. Delay variance in speech renders it unintelligible. The effect of delay variance on video is jerky rendering of frames, a degradation which can be sometimes tolerated. If the underlying network is free of delay variation, the receiving application can simply play out audio as soon as it received. This is however rare in today's packet networks.

The network introduced delay variance is overcome by queuing the data in a buffer, and playing it at a fixed delay at some playback point. Any data that arrives at the receiver end before the playback point can be used to reconstruct the signal. Data arriving after its playback point is useless and discarded.

Similar actions are taken for handling of inter-stream delay variance. If the packets of one stream arrive before the packets of the related stream, a mechanism is needed which synchronizes the two streams, so that they are both played at same fixed delay at a designated playback point.

Playback realtime applications have several service requirements. Since there is often realtime interaction between the sender and receiver, the application is sensitive to data delivery delay, the lower the delay the better. In addition the application has to have some information about the bounds of the delay of the packets in order to set the playback point correctly. The playback applications can often tolerate the loss of some fraction of the packets. Therefore there is no need to delay the playback point so that absolutely all packets arrive beforehand.

Three different types of synchronization can be identified:

1) *playout synchronization*

The receiving application plays out the medium at a fixed time after it was generated at the source thus keeping the end-to-end delay fixed and removing network caused delay variance. Playout synchronization is sometimes referred to as intra-stream synchronization or even intra-media synchronization.

2) *intra-media synchronization*

All receivers play the same segment of medium at the same time. This may be needed in simulations and gaming.

3) *inter-media synchronization*

The timing between several media is restored at the receiving end. An example of this is the synchronization between audio and video (lip-sync) in a video conferencing application

Playout synchronization is the most important type of synchronization. Intra- and inter-media synchronization may be needed depending on the application. A playout unit is a group of packets sharing the same timestamp. A synchronization unit consists of one or more playout units that have a common fixed delay between generation and playout. Examples of synchronization units are talkspurts and video frames.

4.1. Synchronization concepts

As we mentioned above, a receiving application typically buffers packets and delays their playout in order to compensate for variable network delays. The playout delay can be constant throughout the session or it can be adaptively adjusted during session. Depending on the quality, the constant delay approach can be good or bad. In a campus network or a LAN the delay fluctuations are not so significant. However, over the Internet adaptive playout adjustment is currently a necessity. Adaptive playout can be either per-talkspurt or per-packet based. In the former playout delay is kept constant throughout a talkspurt, but is allowed to be different between talkspurts. This results in what is called “compression and elongation of silence periods”: the natural silence periods between talkspurts change. This phenomena is not noticeable if the changes are small enough. The per-packet adjustment introduces gaps in speech, and is not recommended.

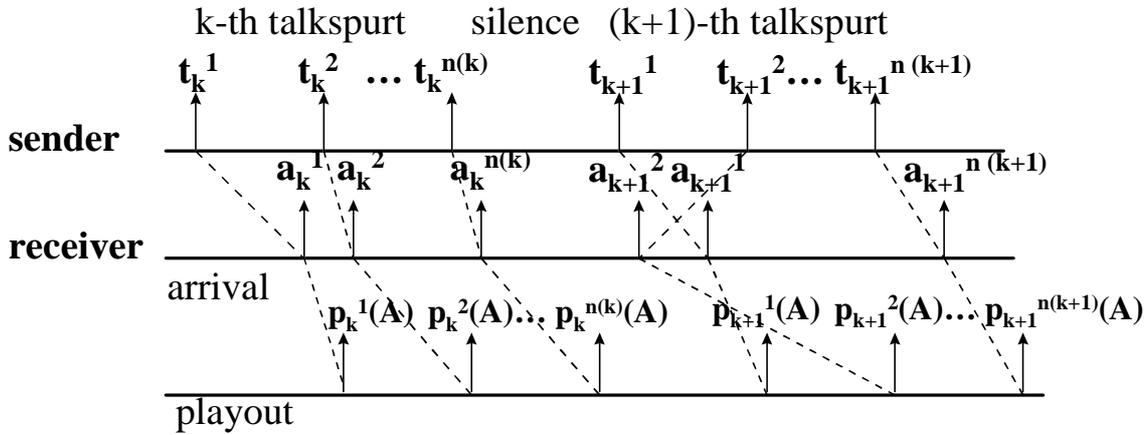


Figure 4-1: Adaptive synchronization.

If, t_k^i : sender timestamp of i -th packet in k -th talkspurt and a_k^i : receiver timestamp of i -th packet in k -th talkspurt then p_k^i : the playout time of i -th packet in k -th talkspurt can be expressed:

$$p_k^i = t_k^i + \text{fixed delay estimate} + \text{variable delay estimate} \tag{11}$$

and the difference in packet spacing is:

$$D(k, k') = (a_k^j - a_k^i) - (t_k^j - t_k^i) \tag{12}$$

D is our starting point for delay variance estimation. The estimate for the interarrival jitter is calculated continuously as each data packet arrives at the receiver. The estimator of interarrival jitter used in RTCP sender and receiver reports uses D to compute J with the formula:

$$J_n = J_{n-1} + 1/16 * (|D(k', k')| - J) \tag{13}$$

Synchronization

The estimate J can be used for QoS adaptation as was mentioned in 3.3.1.

If we look at the synchronization process per synchronization unit based (per talkspurt), we get somewhat simpler definitions for the variables :

a_n is the arrival time of the n :th packet in a synchronization unit

d_n is the delay time of the n :th packet in a synchronization unit

p_n is the playout time of the packet

t_n is the generation time of the packet

o is the fixed delay between the sender and the receiver

d_{max} describes the estimated maximum delay within the network (typically chosen with a 99% confidence interval of $o+d_{max}$)

Given the definitions:

$$a_n = t_n + d_n + o \quad (14)$$

holds for every packet. Also $l_n = p_n - a_n$, is the laxity of packet n . Synchronization methods differ only in how much they delay the first packet of a synchronization unit. All packets within a unit are played out based on the position of the first packet:

$$p_n = p_{n-1} + (t_n - t_{n-1}), \text{ for } n > 1 \quad (15)$$

4.2. Adaptive Playout Delay Estimation

There are three basic adaptive playout delay adjustment algorithms: the *blind-delay*, *absolute timing* and added *variable delay*. Absolute timing maintains a fixed timing relationship between the sender and the receiver. Relative timing ensures that the spacing between packets at the sender and at the receiver is the same measured in terms of the sampling clock.

Blind delay assumes that the first packet in a talkspurt experiences only the fixed delay so that the full d_{max} has to be added to allow for other packets within the talkspurt experiencing more delay, i.e.:

$$p_1 = a_1 + d_{max} \quad (16)$$

The estimate for the variable delay is derived from measurements of the *laxity* l_n . The new estimate after n packets is computed:

$$d_{max,n} = f(l_1 \dots l_n) \quad (17)$$

, where the function f is a suitably chosen smoothing function.

Blind delay only requires an indication of the beginning of a talkspurt. No timestamps are required to determine p_1 . However, timestamps may be required to compute p_n , unless $t_n - t_{n-1}$ is a known constant. In absolute timing the timestamp is used to improve the accuracy of the playout point:

$$p_1 = t_1 + o + d_{max} \quad (18)$$

This is the best estimate. Instead of estimating d_{max} $o + d_{max}$ is estimated as some function of $p_n - t_n$. For this computation, it does not matter whether p and t are measured with clocks sharing a common starting point.

Added variable delay uses a delay accumulator in each packet to which the variable delay experienced in each node is added, thus yielding d_n .

$$p_1 = a_1 - d_1 + d_{max} \quad (19)$$

From (14) we can see that the absolute delay and added variable delay give the same estimate for playout time. The estimate for d_{max} is based on the measurements of d . Given a clock with suitably high resolution, these estimates can be better than those based on the difference between a and p . However using added variable delay would require that all routers be aware of the synchronization protocol. Also determining the residence time within a router may not be feasible. Absolute timing was decided best for the Internet Real Time Protocol RTP. Added variable delay is currently not feasible over the Internet.

4.3. Synchronization Quality of Service

For engineering purposes some rules of thumb are always welcome. For media synchronization these rules of thumb translate quite well into parameters for synchronization QoS. Some parameter values are presented in Table 4-1 The required QoS for synchronization is expressed as the allowed skew. The values presented refer to presentation level synchronization, the values that are reasonable at the user interface. Presentation level sync focuses on human perception of synchronization.

Table 4-1: Quality of Service for Synchronization Purposes.

Synchronization

Media		Mode, Application	QoS
Video	animation	correlated	+/- 120 ms
	audio	lip sync	+/- 80 ms
	image	overlay	+/- 240 ms
		non overlay	+/- 500 ms
	text	non overlay	+/- 240 ms
		overlay	+/- 500 ms
Audio	animation	event correlation	+/- 80 ms
	audio	tightly coupled (stereo)	+/- 11 μ s
		loosely coupled (dialog mode with various participants)	+/- 120 ms
		loosely coupled (background music)	+/- 500 ms
	image	tightly coupled (music with notes)	+/- 5 ms
		loosely coupled (slide show)	+/- 500 ms
	text	text annotation	+/- 240 ms
	pointer	audio relates to showed item	- 500 ms /+750 ms

5. RELATED WORK

In this chapter we will present a selection of the fine research work related to the subject of the thesis. In section 5.1 we give a review of some of the recent Internet measurements, which have shed light to the sometimes unexpected behavior of the Internet. Section 5.2 gives an introduction to the research efforts in trying to make audio more robust to packet loss and the applications adaptive to the state of congestion in the network.

5.1. Internet End-to-End Measurements

A number of measurement based studies have been made with the aim to characterize the end-to-end delay and loss behavior of the Internet [31, 48, 49, 50, 51]. The method of measurement has been to use small probe packets sent at regular time intervals. By varying the time interval between probe packets, it is possible to study the structure of the Internet load over different time scales.

The main conclusions of [31] were:

- the Internet traffic is a mix of bulk traffic (e.g. file transfers) with large packet size and interactive traffic with small packet size;
- the interarrival time of the probe packet stream is consistent with an exponential distribution;
- a compression or clustering phenomenon of the probe packets was noted, i.e. probe packets accumulate behind large Internet packets. Similar compression has been observed in TCP;
- losses of the probe packets are random, when the probe packets use a small fraction of the available bandwidth;
- most losses involve one or two packets and the distribution of the lost packets is approximately geometric.

Some of the observations and conclusions of [48] were:

- IP level service of the network yields high losses, duplicates and reordering of packets;
- the roundtrip transit delay varies significantly even over short periods of time, and is dependent of the time of day;
- there are many sharp rises in roundtrip times;
- in some cases groups of packets were lost before the sharp rise;
- router cache sync causes periodic peaks to roundtrip times;
- multiple packet losses occur with no substantial increase in roundtrip times;
- the losses were not correlated;
- roundtrip time shows step change behavior;
- if the losses were to occur due to buffer overflows only, they must be preceded by and/or followed by an increase in roundtrip time. That does not seem to be the case;
- individual losses cannot be explained on the basis of random bit-errors. Losses may be due to synchronization errors in router software and hardware implementations;
- some of the round-trip times were over a second, which would mean that one second worth of cross traffic were to arrive within 39 ms to servers, which was beyond the input handling capabilities of the processors;
- servers cause large peaks in roundtrip time, by taking “coffee-breaks” of about one second, i.e. during this period all incoming packets are queued, and after the break the packets are processed one at a time.

Somewhat different behavior was observed in [50]:

- the packet delay seen by applications was observed and verified using Whittles estimator to be better modeled as a self similar than as a Poisson process;
- the degree of self-similarity increases with network load and seems to be positively correlated with the packet loss for the path;
- when the degree of self similarity is high the combination of round-trip delay and loss rate may be too extreme for delay- and loss-sensitive applications.

In [49, 51] it was concluded that:

- connections show significant change in characteristics within few seconds;
- losses do not seem be caused just by buffer overflows: there are random losses except when load is high;
- in the earlier experiment [51] a large number of duplicate packets were noted, however in [49] there were nearly no duplicates;
- reordering of packets were not correlated with losses;
- the loss rates were high, from 0.6% to 23%;

Related Work

- the variability of the transit times had decreased in [49]. Minimums were increasing and maximums decreasing.

Measurements were used in [52] to understand the packet loss and delay characteristics of the Internet in relation to its ability to carry interactive voice traffic. Taken that delay variance can be compensated at the receiver, and that one to two lost packets can be restored at the receiver without affecting the perceived quality, a quality measure is developed. The *quality of a channel* is defined as *the fraction of the time that the signal is received without distortion for intervals of time that are “long enough” to carry useful speech segments*. The conclusions were:

- one-two lost packets should be restored when sending 20 ms voice packets over the Internet;
- when more than one packet is lost there are likely to be a large number of lost packets and restoring more than two consecutive packets is not worthwhile;
- the quality of the intrastate connections was better than that of the interstate.

In addition to the fore mentioned experiments in the Internet there are many ongoing activities in Internet performance and workload measurements. These efforts concentrate in collection, analysis and presentation of Internet statistics. One of the leaders of these activities is the National Laboratory for Applied Networking Research (NLANR) [53].

5.2. Redundant Audio and Dynamic QoS Control

A novel of mechanisms has been developed for audio packet loss recovery [40, 41] within the MICE project [54]. With every packet n coded for example with PCM A-law encoding, is included a redundant version of packet $n-1$ coded with a low bit-rate coder. Thus, the audio output at the receiver consists of both PCM- and LPC-coded speech. The redundancy adds a little overhead (24 bytes to a 320-byte PCM encoded packet). The scheme can be extended to recovering from consecutive losses as well as by e.g. including with every packet n the speech frames of packets $n-1$ and $n-2$, or $n-1$, $n-2$ and $n-3$, or $n-1$ and $n-3$, etc. The scheme was noted to work best when the network load is low and packet losses are random.

In [41, 55, 56, 57, 58] are presented schemes that use RTP for dynamic QoS control. The QoS control is used for adapting the send rate and playout delay of conferencing applications to the state of congestion in the network. The state is monitored using RTCP receiver/sender reports and RTP packet interarrival jitter and packet losses. The congestion might be caused by the audio connection, or other connections sharing the network, or both. Typically the source reacts to congestion by decreasing bandwidth requirements. However, this approach has problems. Not all connections react to congestion in the same way, and decreasing the bandwidth does not necessarily result in a decrease in loss rate.

A better approach is to control both the source send rate and the amount of redundant information (FEC) used to minimize the perceived loss in the destination[41]. To achieve this following steps are needed:

- RTCP analysis. The receiver reports of all receivers are analyzed and statistics of packet loss, packet delay variance and roundtrip time are computed.
- Network state estimation. The actual network congestion seen by every receiver is determined to decide whether to increase, hold or decrease the bandwidth requirements and amount of redundancy used. In a multicast conference the decision on the state of congestion is somewhat complicated and requires some statistical methods.
- Bandwidth and redundancy adjustment. The bandwidth and the amount of redundancy of the packets of the multimedia application is adjusted. The bandwidth can be reduced for example by changing the coding scheme or by dropping video frames and even media.

An important function of the receiver application is to adjust the playout delay. Adaptive playout delay algorithms are presented and compared in [58].

6. MEASUREMENTS AND RESULTS

In this chapter we will present measurements of Voice over IP in three different environments: an Ethernet LAN, an IP forwarding campus network and an IP switching campus network. The chapter is divided in four: in the background section, that explains the motivation of the measurements, in the calibration measurement, that was made to see what kind of results we get, in the VoIP over IP forwarding measurement, and in the VoIP over IP switching measurement.

6.1. Background

In this motivation and background section of the measurement part of the thesis we will first explain what we wanted to achieve with the measurements and how we intended to achieve the goals.

6.1.1. The Goals of the Measurements

The goals of the measurements were four-fold:

1. To gain understanding of the delay behavior of IP Voice;
2. To see what is the impact of IP switching on the delay variance of VoIP packets;
3. To measure the total end-to-end delay including the processing delays at the workstations;
4. To test the interarrival jitter estimation of RTCP in a new environment: IP Switching.

6.1.2. Choosing the Measurement Environments

We were primarily interested in delay characteristics of IP voice. Thus a steady source of IP voice packets was needed. We wanted to both measure typical system performance (goal 3), and to understand delay behavior (related to goals 1,2 and 4). Thus a stable IP voice platform with enough flexibility for measurement purposes was needed. We could have created the voice packets also with a packet dumper/generator (such as TCPdump), for all measurements that seek to find out network properties, but we chose to use an actual IP Voice client for all packet creation purposes, to get simultaneous results for both network and end-to-end measurements.

There is a wealth of good, free IP voice software written for Sun Solaris. Sun workstations also have a very accurate timer, which was also important for us. We also have good working knowledge of the platform. Therefore Sun Ultra workstations were chosen as our measurement platform.

We tested a number of IP voice client software, including RAT (Robust Audio Tool), NeVot (Network Voice Terminal) and FreePhone. For our purposes we needed a program that had good configuration and debugging possibilities. Parameters like the playout delay added at the receiving terminal needed to be user configurable. Received RTP packets can be recorded with programs like RTPdump. NeVot has a debugging option that writes the incoming packet headers in a file. The received packet headers are needed for network delay variance calculations. In our tests we found that NeVot was the most versatile when it comes to parameter tuning and packet recording, so we chose it for our measurements.

We were interested in IP voice delay and delay variance characteristics. These parameters are strongly affected by the type and amount of other traffic in the network. Therefore we needed ways to load the network. The load applied to the network needs to be flexibly adjustable, and it needs to reflect actual network traffic. The first requirement can be easily fulfilled using special equipment such as the Adtechs AX/4000 for ATM-environment or Radcom Prism for LANs and WANs. The later requirement can be fulfilled by dumping real production network traffic to a file with sniffer program such as TCPdump and then dispatching the dumpfile to the measurement network. A third way is to run programs like flood ping that generate lots of easily adjustable load in to the network. The weakness of this approach is that the generated traffic does not represent any particular traffic profile. We chose to use a combination of traffic generation. We wanted to test the effect of different background traffic packet sizes and have accurate control of the traffic generation process, so for these measurements real production network traffic was not used as such. The AX/4000 was not used because it is ATM-specific and currently does not support IP over ATM traffic generation.

The network environment for IP Voice measurements we were interested was a campus network implemented with IP switching, which has been proposed as an alternative to the traditional store and forward routing. With the use of cut through switching for flows IP switching is supposed to help relieve the router bottle-neck of the current IP-based intranets and the Internet. We were interested in the suitability of this technology for improving the quality of IP voice. The question we hoped to answer with the measurements was: does IP switching improve the quality of VoIP, and if so is it a viable option for networks that need to be IP voice aware, in the sense that they give better QoS for VoIP than the traditional networks. To have some basis for comparison of the results of the IP forwarding and switching results we decided to measure VoIP over Ethernet.

6.2. Calibration Measurement - VoIP over Ethernet

This section handles our calibration measurement in Ethernet. We will first go through the laboratory setup and explains what was measured exactly, and how it was done. We will also present the results of the measurement. We measured both the end-to-end delay of Voice over IP over Ethernet and the packet spacing of voice streams over a non-loaded network. We also performed measurements with the network loaded with different background traffic.

6.2.1. Setup of the Measurement

In the first measurement we had two Sun Ultras connected to a 10BaseT Ethernet. On both workstations we had VoIP clients, NeVot (Network Voice Terminal) version 3.34 and Sun standard audio hardware. To the workstations we attached an external measurement system that can measure the delay of an analog pulse. For generating pulses we had a standard function generator, that was attached to both the delay measuring device and the workstation. The receiving workstations audio output was attached to the delay measuring device. In addition to this using the debugging option of Nevot the timestamps of incoming audio packets at the receiver were recorded. The timestamps can be used for calculating the packet spacing difference of a pair of packets at the receiver compared to that at the sender. From the packet spacing differences we can for example calculate network caused packet interarrival jitter of the audio stream.

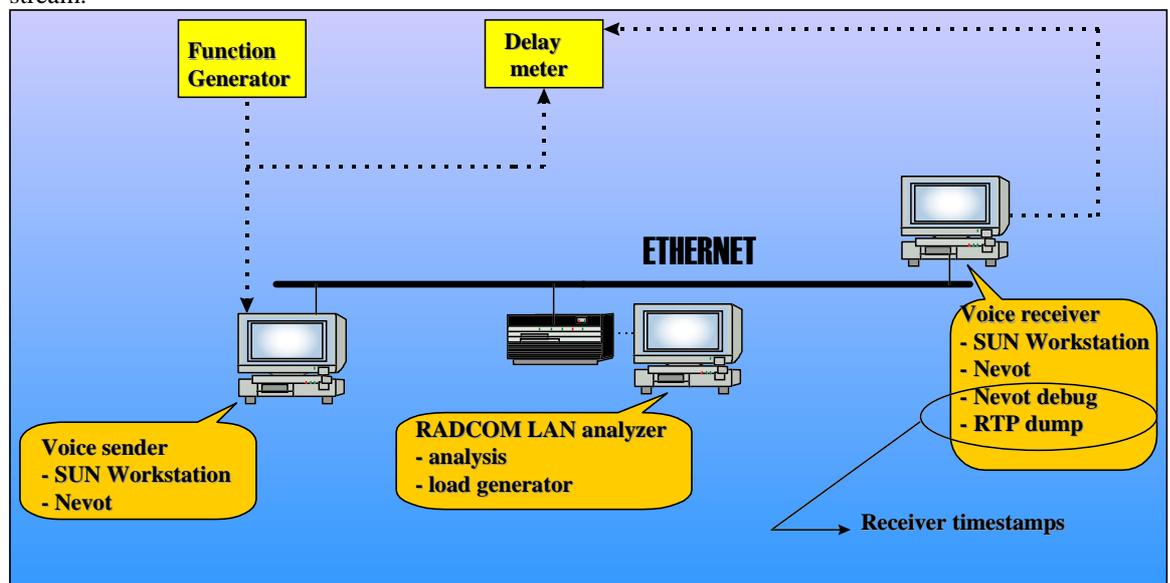


Figure 6-1: Setup of the VoIP over ethernet measurement.

6.2.2. Measurement of the total delay

The total end-to-end delay from the voice sender's microphone to the receiver's speaker was measured using a method called pulse delay measurement. A one kHz sine pulse was generated with a function generator exciting the generator by applying an appropriate dc current into the input of the generator. The pulse was fed through an analog audio mixer into the line input of the sending Sun Ultra workstation and to the A-input of the Hewlett Packard 5300A digital timer/counter. The pulse is sampled, coded and packetized in the workstation, after which it is sent through the network. The receiver de-packetizes, decodes and d/a-converts the pulse and feeds it to the output. The receivers output is attached to the B-input of the counter.

When the pulse fed into the A-input of the HP 5300A is higher in voltage than the set threshold a counter starts. The counter stops when the same happens to channel B. The resolution of the counter is $0,1\mu S$. The adjustment of the threshold must be exact and the input voltages sufficiently high in order to receive accurate results. If the input voltages are too low the thresholds are too close to zero and the results may be wrong and lost in noise.

First we wanted to measure the hardware caused delays in the sending end, in the workstation. For this the setup was as explained except that the B-input of the HP 5300A was attached to the line output of the sender. We then using a program called TK_Audio using the monitor option directed the line input to the line output. The input signal was A/D-converted and then D/A converted in the audio hardware. The samples did not go to the VoIP program for coding, and thus no buffering and coding delays were introduced. The delay was measured a number of times and the average is in Table 6-1, on the row HW delay. The delay was between 7,77 and 12,85 ms, with an average of 8,9 ms. No time or workstation load dependency was noted.

In the next measurement we wanted to find out the total delay in the workstation including the buffering and coding delays of the VoIP client software. The software used was Nevot (Network Voice Terminal) version 3.34. In the software the packet length was adjusted to 20 ms. This means that the input was sampled at 8000 Hz (one sample = 125 μ S), and that a 20 ms block of these samples were coded and packetized at a time. The program must therefore buffer the samples in memory before coding. In Nevot we used the monitor option so that the coded samples were decoded and then D/A-converted and fed into the output. A strange phenomena was noted. When the program is first started the coding delay can be as low as 45 to 67 ms, but then after some ten seconds the delay starts to gradually rise to the point where it is over 100 ms. After this some kind of steady-state is observed and the delay varies between 100 and 120 ms. The average is in Table 6-1. Subtracting the coding and decoding sample lengths of 2 x 20 ms and the hardware delay of around 9 ms gives a buffering and processing delay of 0 to 54 ms, where the latter figure is of the order seen by the VoIP user during a call. It should be noted that no playout delay was inserted.

Table 6-1: Terminal delays.

	Delay (ms)
HW delay	8.9
VoIP Client	103.9

Next we proceeded into measuring the delay of the VoIP call over the Ethernet network. The network was kept lightly loaded in order to observe only the fixed component of the network delay in addition to the terminal coding, decoding, packetization, buffering and processing delays on both ends. The program used was again Nevot. No playout delay was added at the receiver. The same time dependency as seen in the previous measurement was noted. When the program and/or the receiving was started the total end-to-end delay varied around 58 to 68 ms, and the delay gradually climbed over 100 ms. The steady-state delay was from 101 ms to 110 ms, with an average of 104.5 ms. This behavior was seen in repeated measurements. The same delay was also measured using a packet length of 200 ms. The difference was notable. some 200 ms higher. Setting the playout delay option on at the receiver adds delay according to the playout delay settings. The default variance multipliers and initial playout delay options are so high that the end-to-end delay was over 500 ms. Even the minimum setting tend to give an additional 50-100 ms of playout delay, even if the network is a lightly loaded LAN. Setting the playout delay off did not result in degradation of the quality of the speech in our unofficial subjective tests, but setting the playout delay on resulted in the delay becoming annoyingly high.

Table 6-2: End-to-end delays over Ethernet.

Packet length 20 ms	104.5 ms
Packet length 200 ms	294.8 ms

6.2.3. End-to-End Delay of PC

Currently the most popular Voice over IP platform is a Pentium PC running Windows'95 and one of the VoIP clients. The interoperable H.323 clients such as Microsoft Netmeeting, Intel Internet Video Phone and Vocaltec Internet Phone are increasingly popular. Therefore to get some-hands on information on the current VoIP operating environment we decided to measure a typical PC system.

The measurement system was as follows. We had two Pentium PCs running Windows'95 and Microsoft Netmeeting. Details of the PCs are in Table 6-3. The PCs were both in the same ATM emulated LAN. The ATM switch used was a ForeRunner ASX-200WG. We measured the end-to-end delay in the same way as in 6.2.2. The results are summarized in Table 6-4.

Table 6-3: The PC details.

	Sending PC	Receiving PC
Processor	Intel Pentium 133MHz	Intel Pentium 100 MHz
Memory	32 MB	32 MB
Cache	Pipelined Burst 256 Kbytes	Pipelined Burst 256 Kbytes
Soundcard	Soundblaster 32 AWE	Soundblaster 16 PnP
Drivers	DirectX 5.0 (full duplex)	DirectX 5.0 (full duplex)
OS	Windows'95	Windows'95
VoIP client	NetMeeting 2.1 Beta	NetMeeting 2.1 Beta
Audio coding	G.711 (CCITT u-law)	G.711 (CCITT u-law)
Network Card	Collage 25 ATM	ForeRunner LE 25Mbps ATM

Table 6-4: Terminal and end-to-end delays.

Terminal hardware delay	5,4 ms
Network delay (ping)	<1 ms

End-to-end delay over LANE	190 ms
----------------------------	--------

The PC system showed a very large standard deviation of 20 ms for the end-to-end delays. The terminal delay which only includes sampling and playout at the soundcard showed very consistent performance with a standard deviation of less than one ms. The network delay measured was under 1 ms, which is below the resolution of the PC timer. The copying of the samples to memory, buffering, coding and IP stack processing at both ends cause most of the end-to-end delay in this measurement. More advanced coding methods such as the G.723.1 seemed to increase the delay.

6.2.4. Measurement of the Delay Variance of the Voice Packets

In this measurement we had two Sun Ultras connected to a 10BaseT Ethernet. On both workstations we had VoIP clients, NeVot (Network Voice Terminal) version 3.34 and Sun standard audio hardware. We had a constant source of audio fed to the coder, and silence detection was not in use. We sent voice packets at a constant net bitrate of 64 Kbps, one packet every 20 ms. The sampling rate was 8000 Hz, one sample every 125 μ S. At the receiver using the debugging option of Nevot the timestamps of incoming audio packets were recorded.

At first the network was loaded only with the measured audio stream. From the recorded packet timestamps packet spacing difference is calculated:

$$D(i, j) = (a_j - a_i) - (t_j - t_i) \quad (6.1)$$

where,

$D(i,j)$: difference in packet spacing at the receiver compared to the sender for a pair of packets i and j

t_i : sender timestamp of i -th packet

t_j : sender timestamp of j -th packet

a_i : receiver timestamp of i -th packet

a_j : receiver timestamp of j -th packet

The results of Table 6-5 we obtained by calculating $D(1,2)$ to $D(n-1, n)$ for our measurement and then applying basic statistical methods. The mean deviation of the Packet spacing difference used by the terminal programs for playout delay adaptation is negligible.

Table 6-5: VoIP over Ethernet with no background load.

	Average Packet spacing difference (ms)	Mean deviation (ms)	Variance of Packet spacing difference (ms)	Standard deviation (ms)	Maximum Packet spacing difference (ms)	Minimum Packet spacing difference (ms)
No load	-0.0034	0.0822	1.904	1.380	58.76	-19.88

Next we wanted applied additional traffic to the network to see the effect it has on the voice stream. We had three traffic scenarios: one where the LAN traffic is mostly small realtime packets of constant frame rate, one where our voice stream has to compete for transmission media with large nearly maximum sized packets of constant frame rate and one where the traffic is variable both in frame rate and packet size. The traffic was generated using Radcom Prism 200 protocol analyzer. The network traffic summaries are in Table 6-6:Traffic summary. Again we calculated the Packet spacing differences of the voice stream. The results are summarized in Table 6-7.

Table 6-6:Traffic summary.

Traffic description	Load (Mbps)	Frames/s	Packet size (bytes)
Small Packets	2.9 - 3.1	2000	160
Large packets	6.8 - 7.2	590 - 630	1450
Variable Bursty	0.6 - 6	10 - 900	160 - 1450

Table 6-7:VoIP over Ethernet with different background loads.

Load type	Average D (ms)	Mean deviation of D (ms)	Variance of D (ms)	Standard deviation of D (ms)	Maximum D (ms)	Minimum D (ms)
Small packets	-0.0526	2.346	9.577	91.66	168.1	-159.6
Large packets	-0.0829	32.83	40.44	1635	119.4	-119.2
Variable packets. bursty load	-0.0526	3.289	13.18	173.5	183.8	-130.0

6.2.5. Analysis and Conclusions of the Ethernet Measurements

The whole idea of the Ethernet measurements was to evaluate both our voice platform and our measurements methods in an environment from which exists a firm ground of experimental and theoretical knowledge. Therefore before proceeding to the measurements in IP switching network, we needed to thoroughly analyze the Ethernet measurements.

For analysis we decided to calculate the packet spacing difference distributions or frequencies and packet spacing difference percentiles for all our Ethernet data. These are depicted in Figures 6-2 to 6-10. First looking at the measurements made in the non-loaded network (depicted in Figure 6-2 and Figure 6-2) we see that the frequencies have a center around 0.001 ms and a very large portion of the packet spacing differences is less than 0.03 ms. The percentiles of *D* show that 99% of all packet arrive with the same packet spacing as they had at the sender (20 ms packetization interval in our experiments). This in fact means that no delay adaptation is necessary.

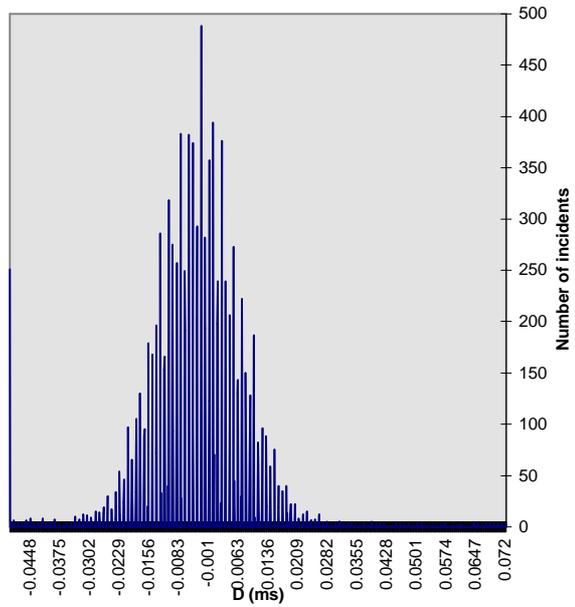


Figure 6-2: Packet spacing difference distribution (frequencies) of VoIP over non-loaded Ethernet.

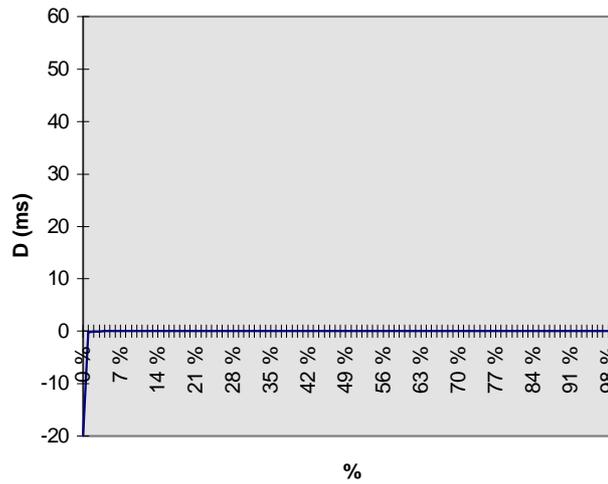


Figure 6-3: Packet spacing difference percentiles of VoIP over non-loaded Ethernet.

In Figure 6-4 is depicted the frequencies of the packet spacing differences for the measurement where we loaded the network with fixed sized 160 byte packets. In this measurement (depicted in Figure 6-5) again most packets (88%) arrived with $D = 0$. Despite this we observe widening of the distribution and the point of 99% arrivals is around 20ms. Whether playout delay adaptation or fixed playout delay is more advantageous in this case can be disputed. Some sort of redundancy [40], could provide better quality for the user than added playout delay. An interesting point to notice is the three peaks in the distribution.

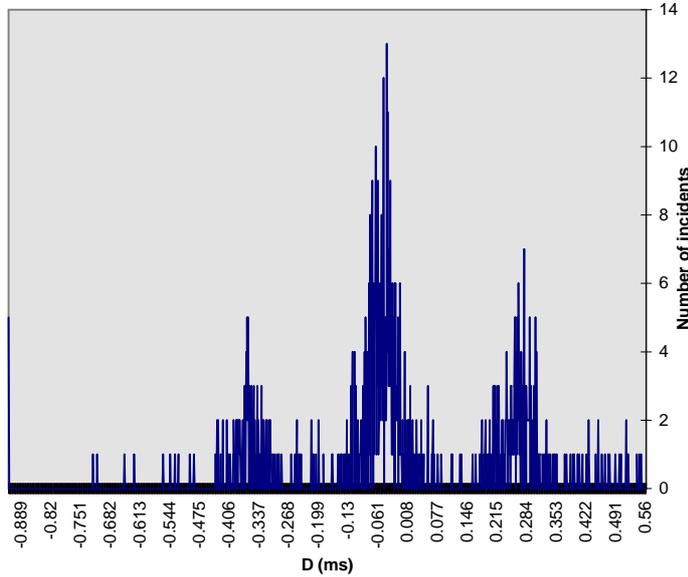


Figure 6-4: Packet spacing difference distribution (frequencies) of VoIP over Ethernet loaded with small packets.

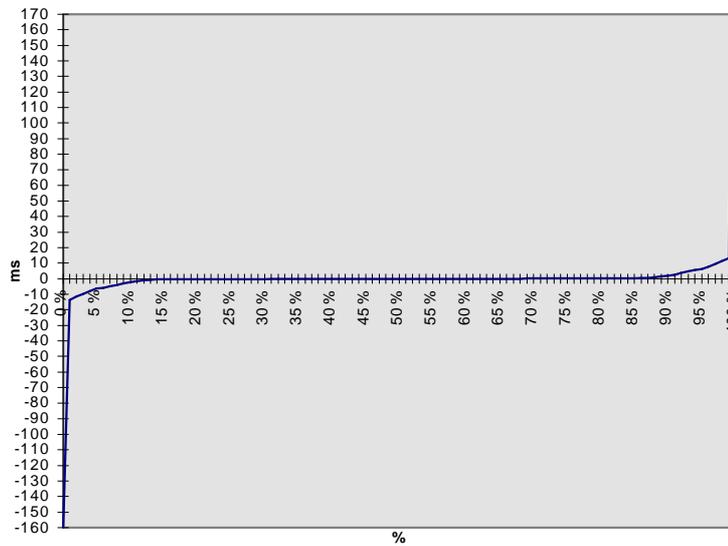


Figure 6-5: Packet spacing difference percentiles of VoIP over Ethernet loaded with small packets.

The bursty load and variable packet size (more or less a bimodal small and large packet length distribution) measurement frequencies of D and percentiles of D are depicted in Figure 6-7 to Figure 6-10. To observe the same three peak distribution as for the small packets we have also included Figure 6-7 with larger intervals on the x-axis. The percentiles of D show that for our quite highly loaded LAN with bursty traffic the D of 99 % is quite high, around 50 ms, when at the same time $D = 0$ is 92%.

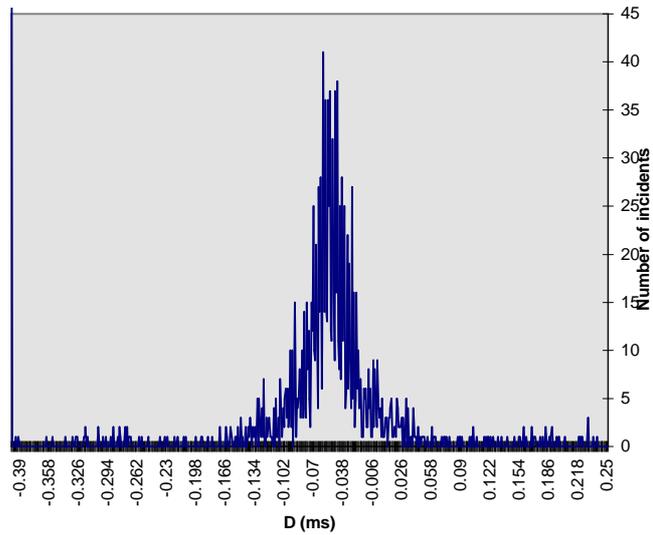


Figure 6-6: Packet spacing difference frequencies of VoIP over Ethernet with a bursty load.

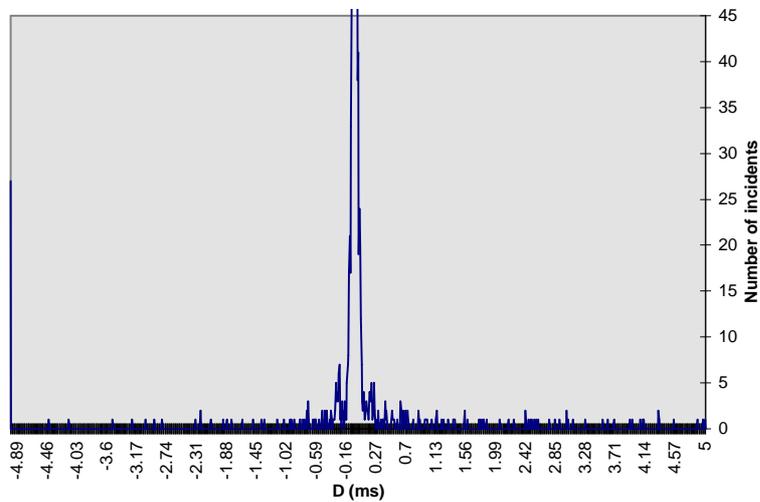


Figure 6-7: Packet spacing difference frequencies of VoIP over Ethernet with a bursty load (different scaling).

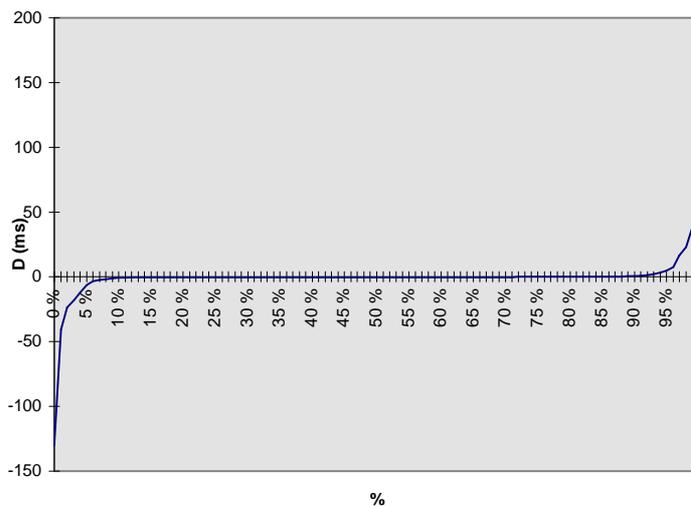


Figure 6-8: Packet spacing difference percentiles of VoIP over Ethernet with a bursty load.

For last measurement in Ethernet we tried to apply as high load as we could to the network. The packets generated were fairly large, 1450 bytes. The distribution of D depicted in Figure 6-9 is like white noise in nature having a spike around zero. The percentiles of D tell a tale of very highly overloaded LAN

connection. The D of zero is at 57th percentile, with a more or less linear rise up to the point (95%, 60 ms) where we observe an exponential rise in D to the largest value of D .

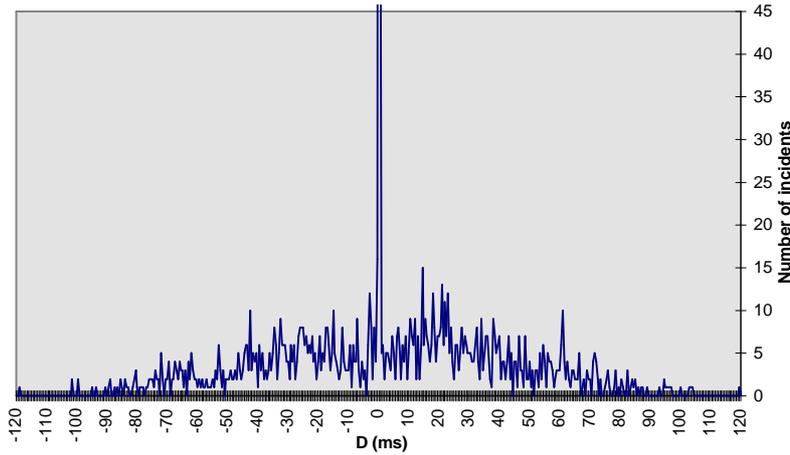


Figure 6-9: Packet spacing difference frequencies of VoIP over Ethernet loaded with large packets.

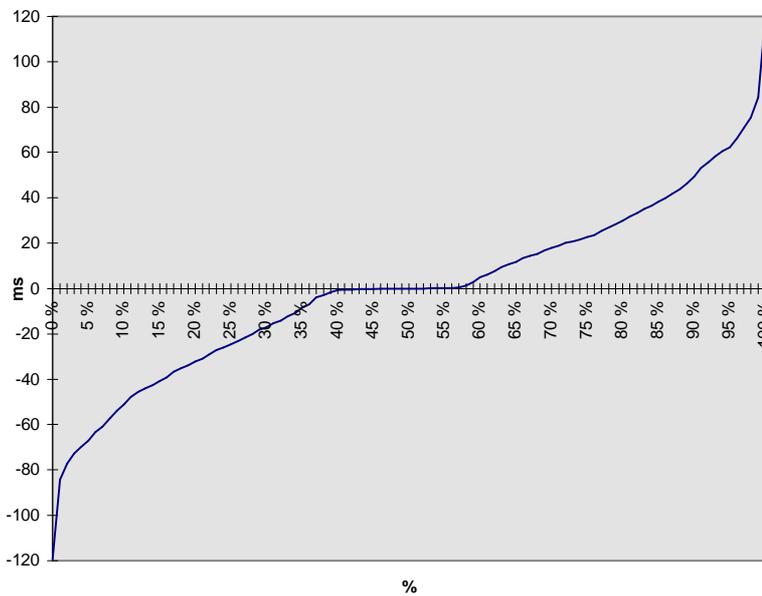


Figure 6-10: Packet spacing difference percentiles of VoIP over Ethernet loaded with large packets.

As a conclusion of the VoIP over Ethernet measurements the methods used seem to catch the interesting phenomena in the network. If we look back to the Table 6-7, where we have a summary of the experiments we see that the largest mean deviation of the VoIP packet spacing differences is that of the experiment with large packet size load. This translated into a distribution of D that is like white noise with a spike in 0. This is clearly worst case behavior. Thus it would make sense to measure VoIP over IP switching with same methods and also to use large packet sizes.

6.3. Measurement II: VoIP over Packet Forwarding Network

This section handles the Voice over IP forwarding campus network. We will first explain the measurement setup: what was measured and how it was done. We will also present a statistical summary of the results.

6.3.1. Setup of the Measurement

In this measurement we studied an IP voice connection over an IP switching network that was set to packet store and forward mode. In other words the network did not make any cut through connections over the ATM switch fabric. All packets including our long lived IP voice flow were processed in the conventional IP router way. The sending voice terminal was connected with a 100BaseT Ethernet to an IP Switch Gateway router. From there the packets were routed and forwarded through an ATM link running on OC-3 to an ATM switch through the default VP/VC. From there the packets go through the default VP/VC to the IP switch processor. The IP switch processor routed and forward the packets to the neighboring IP switch processor ATM switch pair. There the switch processor routes and forwards the

packets to the next IP switch, which is an IP switch gateway. Our receiving voice terminal was connected to this gateway using 100BaseT. On the receiving terminal we were again recording packet timestamps.

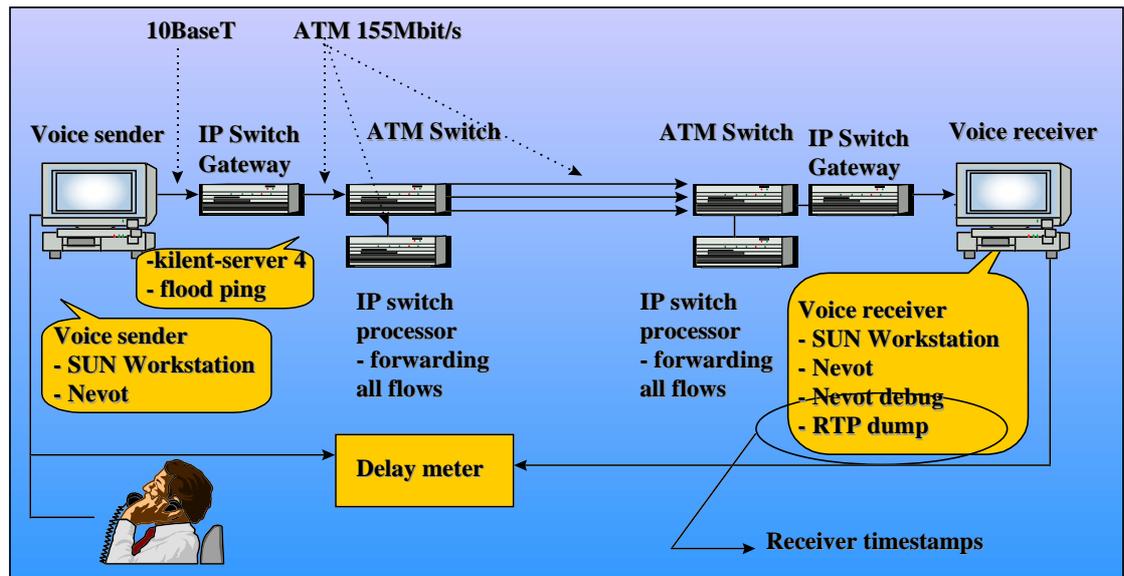


Figure 6-11: Setup of the VoIP over IP forwarding measurement.

6.3.2. Measurement of the Voice Packet Spacing Differences

The measurement proceeded in the same way as in 6.2.4. We first had a VoIP connection through a network with no added traffic. Naturally in a network with routers there is always some router incurred traffic such as OSPF hello-messages.

Next we wanted some understanding of the network behavior under load. We had the assumption that the router functions in the IP switch processors and the border gateways would be the bottlenecks, causing delay variation to the voice stream. We wanted to push the forwarding functions in the switch processors. Caching routes relieves the route table look-up bottleneck in a small campus network like ours, so there was no point in trying to push the loading of that further. We decided to use a more direct approach. Flooding ping is a very effective way to load a network. It also is one of the denial-of-service-type of hacking methods, that has caused networks to go down in the past. As is indicated by the Ethernet measurement and in previous studies large packets cause more delay variance to a voice stream than small packets. Therefore to measure the worst case performance we used 1500 byte packets. We ran the ping processes from the routers themselves in order not to load the LANs in the edges, and really get only the performance difference between IP switching and IP forwarding in the two core switches. The workload on the edge routers and the traffic in the connected LANs is equivalent in the IP switching and IP forwarding case. The measurement approach will be discussed in more detail in section 7.2.

We made several measurements with raw TCP-traffic loading the network and causing our Voice stream delay variance. Using a client-server network traffic generator and throughput measurement application Kilent-Server version 4 written in Perl by Markus Peuhkuri we loaded the network with large, 1500 byte TCP-packets. The client application running on the IP switch gateway sent the packets to the server application running on the other end of the network in an IP switch gateway. When running the client application it starts sending TCP packets between the server and client using a user defined port. The client starts a timer and starts sending packets with MTU size 1500 bytes and packet frame size 1500 bytes.

We experimented with one, two, four and eight sending processes. The average load in the network did not increase notably with the addition of sending processes which implicates that the sending gateways forwarding function was saturated already with two processes. The CPU loads were from 20% to 40% (ps-u) on the gateways, which also supports this.

A summary of the results of the VoIP over IP forwarding measurements are shown in Table 6-8:Summary of IP forwarding measurements., and a summary of the traffic in the network during the measurements is in Table 6-9.

Table 6-8:Summary of IP forwarding measurements.

Load type	Average D (ms)	Mean deviation of D (ms)	Variance of D (ms)	Standard deviation of D (ms)	Median of D	Maximum D (ms)	Minimum D (ms)
No load	-0.00557	0.133	2.56	1.59	-0.006	63.6	-19.7

Measurements and Results

Flood ping	0.20	0.606	6.65	2.58	0.10	74.5	-19.8
K-S 2 processes	-0.01	0.617	7.19	2.68	0.11	105.5	-19.8
K-S 4 processes	-0.01	0.602	5.40	2.32	-0.01	90.2	-19.8
K-S 8 processes	-0.01	0.613	6.92	2.63	0.01	108.6	-19.8

Table 6-9:IP forwarding measurements traffic summary.

Traffic description	Load (Mbps)	Packet size (bytes)	Throughput (Mbps)
Flood ping	2 x 60	1500	-
K-S 1 processes	1 x 80	1500	1 x 73
K-S 2 processes	2 x 41 - 49	1500	2 x 42
K-S 4 processes	2 x 60 - 70	1500	2 x 51
K-S 8 processes	2 x 10 - 90	1500	2 x 52

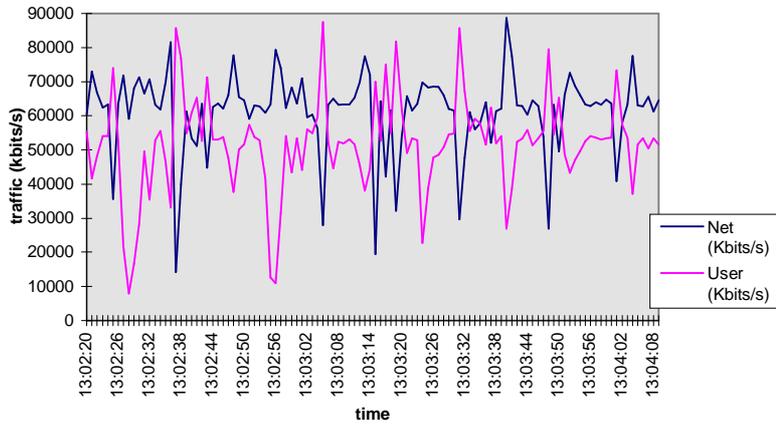


Figure 6-12: Network traffic during measurement K-S 8

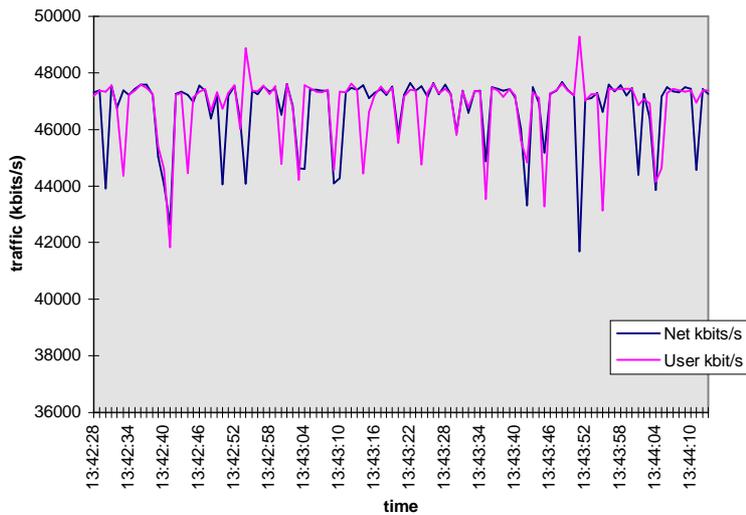


Figure 6-13: Network traffic during measurement K-S 2

6.4. Measurement III : VoIP over IP Switching Network

This section presents the measurements in IP switching network. We begin by describing the measurement setup. After that follow the measurements and a summary of the results..

6.4.1. The Setup of the Measurements

In this measurement we studied an IP voice connection over an IP switching network that was set to establish switched flows for long lived connections ports and IP addresses. All long lived IP packet flows were identified, and cut through connections through the ATM switch fabric were established. The IP switch gateways on the edges were still using conventional store and forward, because cut through connections can only be established between IP switches. The sending voice terminal was connected with a 100BaseT Ethernet to an IP Switch Gateway router. From there the packets were routed and forwarded through an ATM link running on OC-3 to an ATM switch through the connection specific VP/VC. There using ATM cell switching the packets in cells are switched to the correct port on the correct VP/VC. From there the packets in cells go to the neighboring ATM switch input. Cell switching is used and the flows switched to the correct port from where they finally go to the border IP switch gateway. The gateway routes and forwards the packets to the correct LAN. Our receiving voice terminal was connected to this gateway using 100BaseT. On the receiving terminal we were recording packet timestamps.

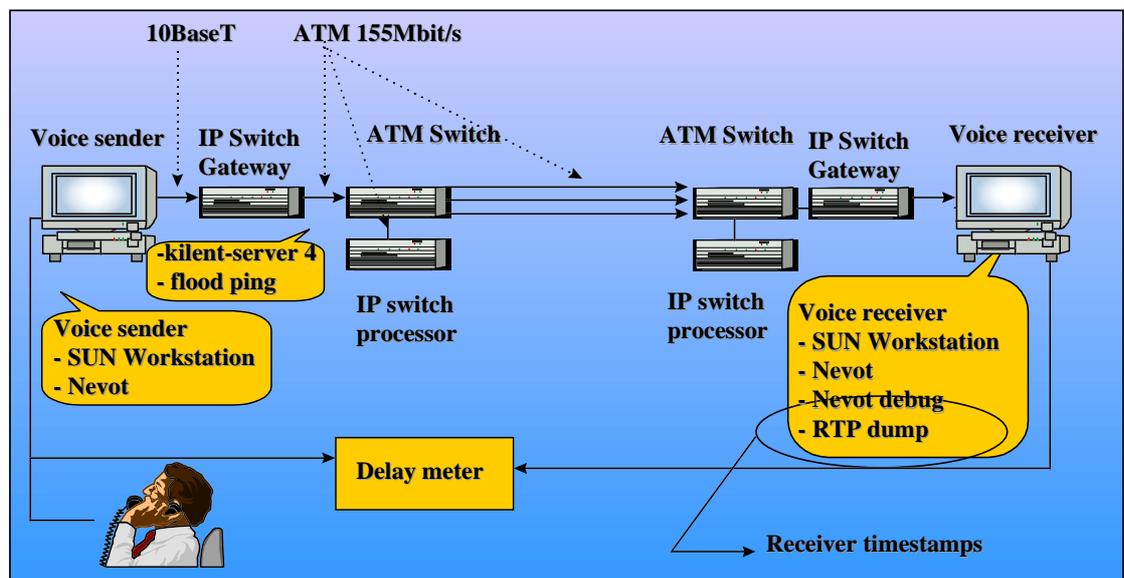


Figure 6-14: Setup of the IP switching measurement.

6.4.2. Measurement of the Delay Variance of the Voice Packets

In this measurement we made a policy on all IP switches that all flows longer than five packets should be switched. The measurements were made in the same way as in 6.3, first with an unloaded network then with flood ping loading the network, continuing with a measurement with our program loading the network. Finally we decided also to try a mixed load with our program and flooding ping at the same time. The flood ping load was approximately 20 Mbps and we were running two sending processes, one on each border gateway.

A summary of the results of the VoIP over IP switching measurements are shown in Table 6-10, and a summary of the traffic in the network during the measurements is in Table 6-11.

Table 6-10: Summary of IP switching measurements

Load type	Average D (ms)	Mean deviation of D (ms)	Variance of D (ms)	Standard deviation of D (ms)	Median of D	Maximum D (ms)	Minimum D (ms)
No load	-0.01	0.110	2.47	1.57	-0.01	66.4	-19.8
Flood ping	-0.01	1.131	9.14	3.02	-0.01	135.7	-19.8
K-S 2 processes	-0.01	0.311	4.94	2.22	-0.01	108.3	-19.3
K-S 4 processes	-0.01	0.376	5.23	2.29	-0.01	102.1	-19.8
K-S 8 processes	-0.01	0.297	5.47	2.34	-0.01	102.7	-19.8

Mixed load	-0.01	0.982	8.56	2.93	-0.01	101.8	-19.8
-------------------	-------	-------	------	------	-------	-------	-------

Table 6-11:IP switching measurements traffic summary.

Traffic description	Load (Mbps)	Packet size (bytes)	Throughput (Mbps)
Flood ping	2 x 60	1500	-
K-S 1 process	1 x 90 - 130	1500	110
K-S 2 processes	2 x 20 - 100	1500	81 & 98
K-S 4 processes	2 x 44- 90	1500	51 & 60
K-S 8 processes	2 x 60 - 70	1500	37 & 56
Mixed Load	2 x 30 -80	1500	37 & 53

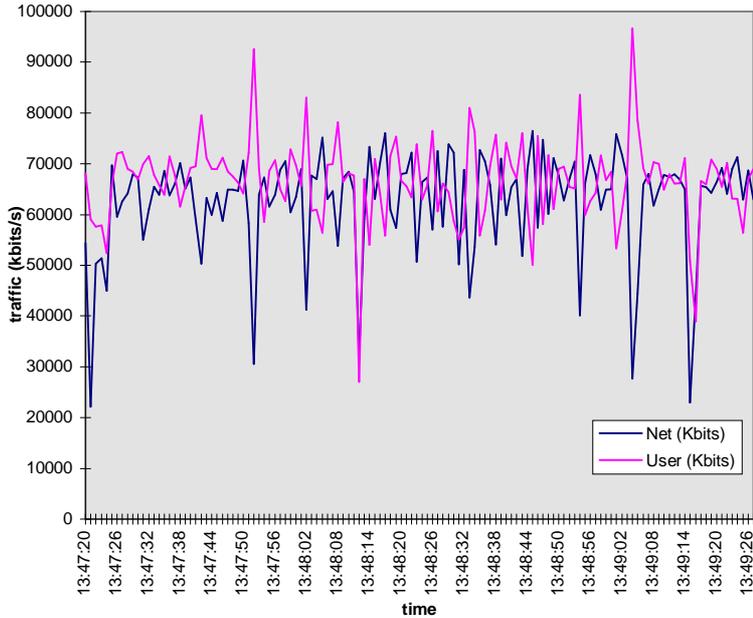


Figure 6-15: Network traffic during measurement K-S 2.

6.4.3. Packet Spacing Differences and Interarrival Jitter Estimates

In this section we study the packet spacing differences in time, and the interarrival jitter estimates. This per packet graphing of the packet spacing difference gives valuable insight to the process. We will also present interarrival jitter estimates with different weighting factors. By varying the weighting factor we want to see how the different weighting factors affect the estimators.

From Figure 6-16 and Figure 6-17 we see that in these instances both in the switched and forwarded case the packet spacing differences vary in the same way. In the forwarded case there is a long period (around 1200 packets) where the average D is biased on the positive side. The switched case has a larger amount of maximums exceeding our largest D of 10 ms on the scale. These peaks give a sharp rise to the estimate of interarrival jitter. We also notice a fluctuation of D between negative and positive values. After a sharp positive peak (packet arrives late) we observe a negative peak where the packets have arrived too soon. This is natural because we are observing the packet space difference between consecutive packets. When a packet arrives late the following packets have accumulated and arrive one after another in a very short time, which is less than the sample length, but because they each contain 20 ms of samples D becomes negative.

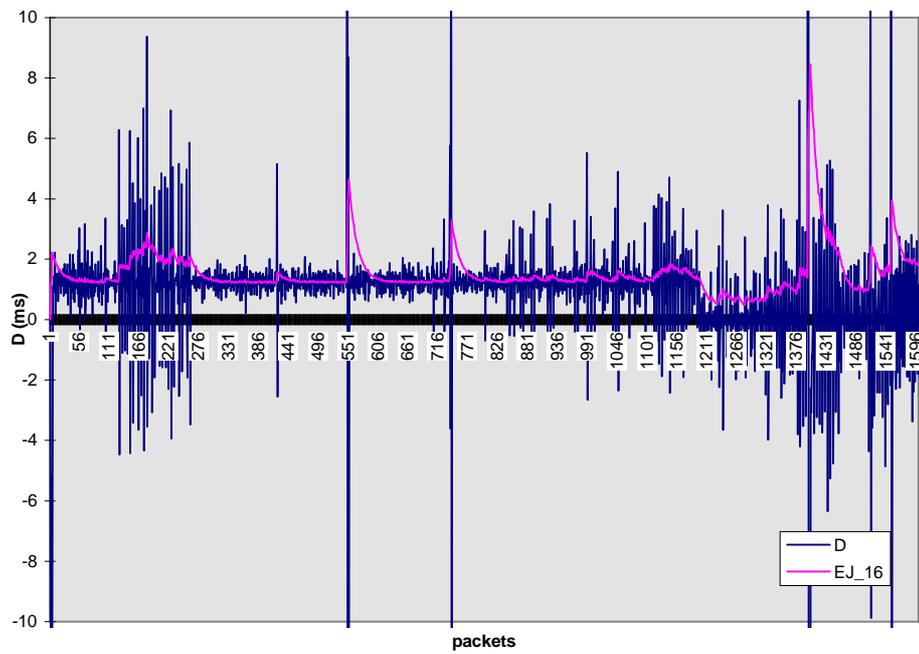


Figure 6-16: Packet spacing differences of IP forwarding with flood ping load.

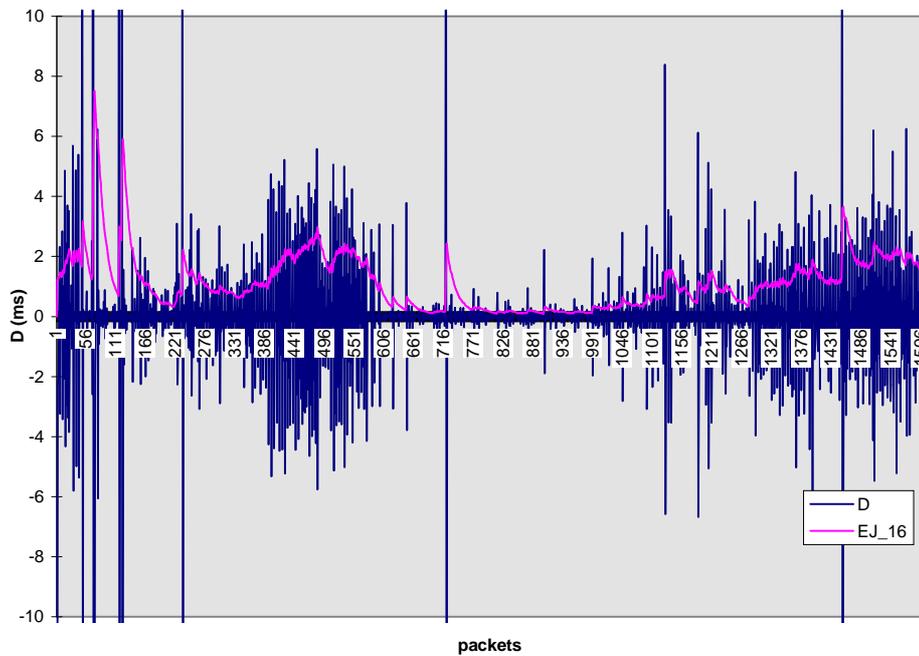


Figure 6-17: Packet spacing differences of IP switching with flood ping load.

If we look at Figure 6-18 and Figure 6-19 we see that in these instances the IP switching has a smaller variance of D , but IP forwarding has less high peaks. The estimate of the interarrival jitter in Figure 6-18 is somewhere around 1 ms and in Figure 6-19 less than 0.5 ms.

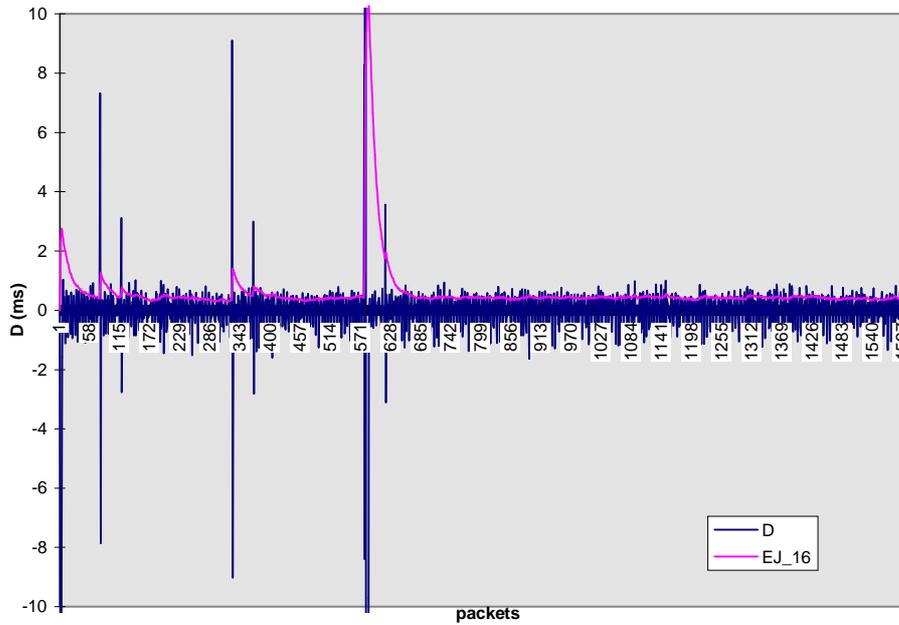


Figure 6-18: Packet spacing differences of IP forwarding with K-S 2 processes.

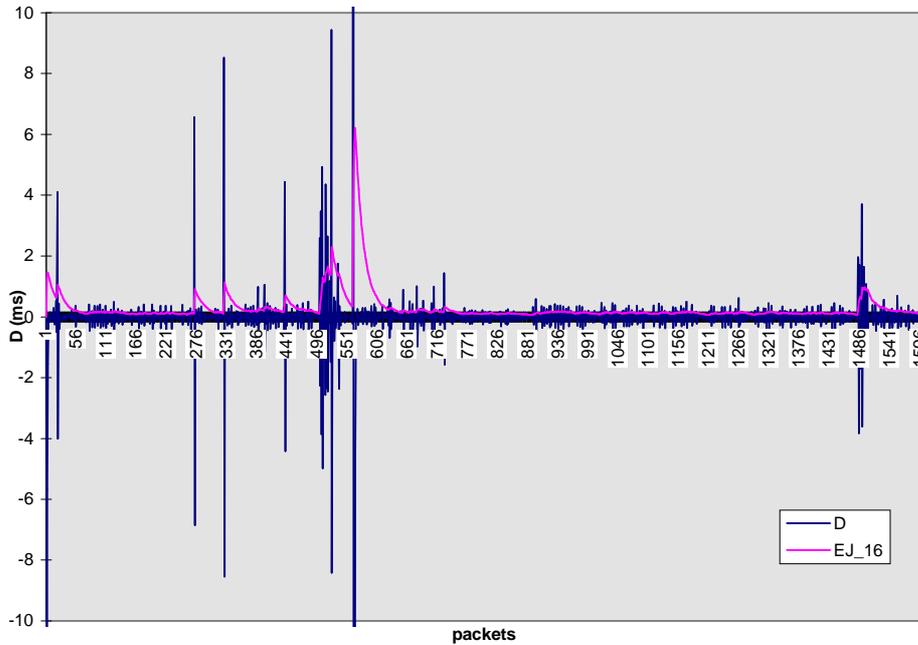


Figure 6-19: Packet spacing differences of IP switching with K-S 2 processes.

In Figure 6-20 and Figure 6-21 the packet spacing difference for 1600 consecutive packets is shown for the case where we loaded IP forwarding and IP switching networks with four instances of our Kilent-Server4 program. The IP switching has still quite low overall D while IP forwarding has a larger deviation of D in average. Notably high peaks in D occur in Figure 6-20 between packets 1040 and 1335. There we can see the rise of the estimate of interarrival jitter to around 4 ms.

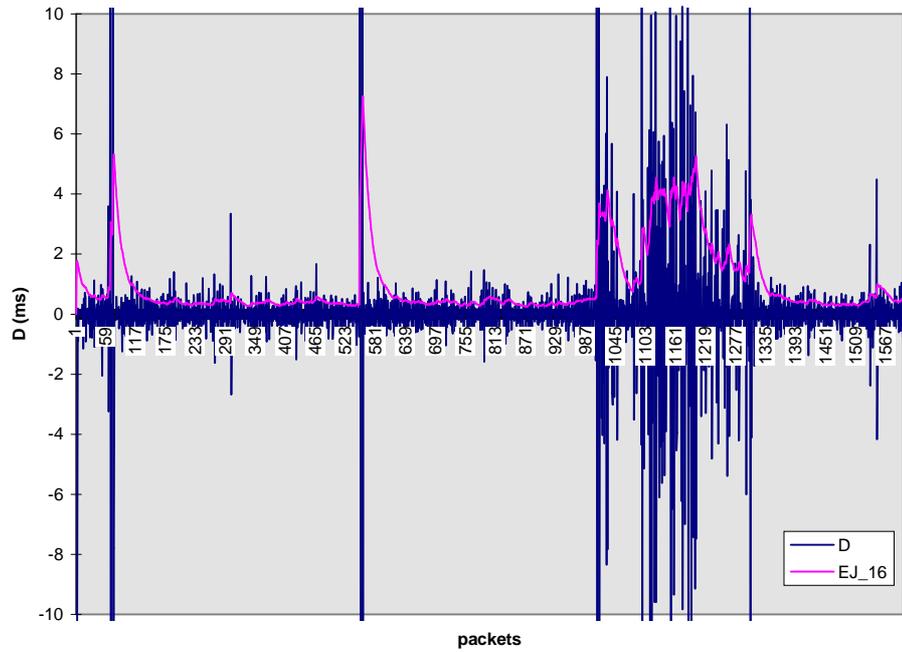


Figure 6-20: Packet spacing differences of IP forwarding with K-S 4 processes.

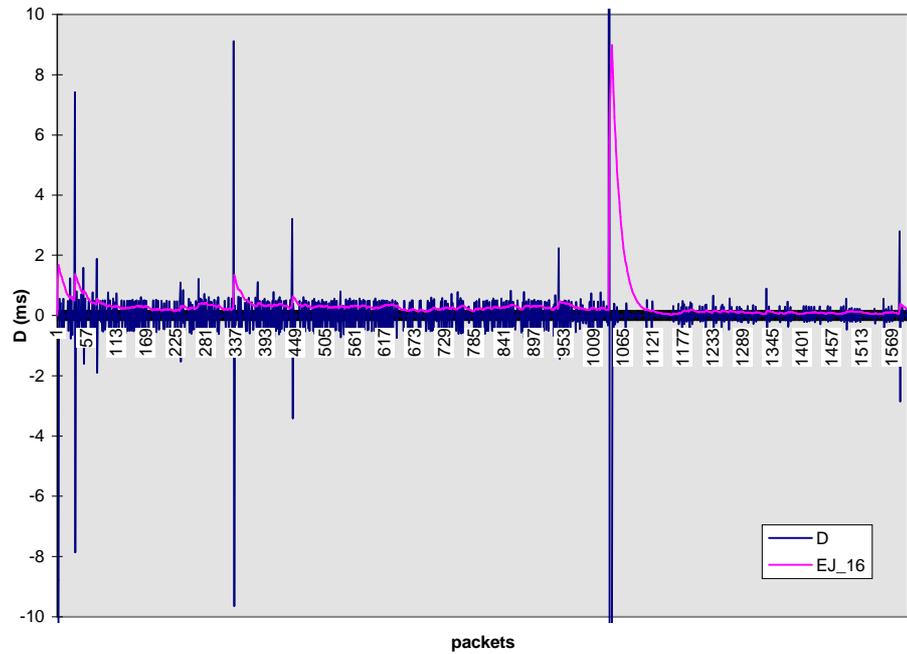


Figure 6-21: Packet spacing differences of IP switching with K-S 4 processes.

In Figure 6-22 we can see the effect of our mixed load on the voice stream over IP switching network. The packet spacing D fluctuates between 4 to 6 ms and -4 to -6 ms. The interarrival jitter estimate follows quite nicely the fluctuations of D .

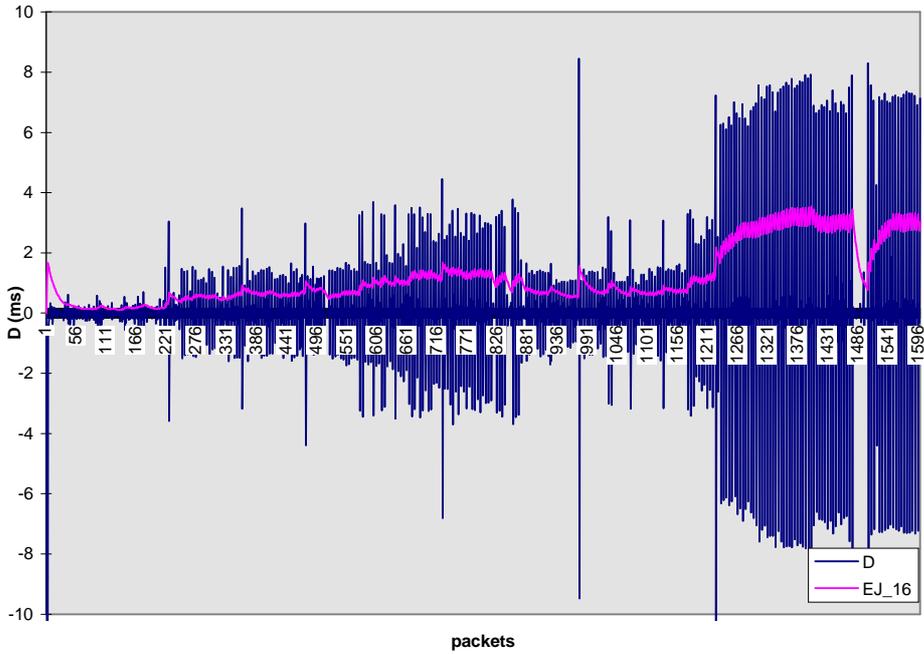


Figure 6-22: Packet spacing differences of IP switching with mixed load.

We computed the errors of the estimates of the interarrival jitter. For every estimate of the interarrival variance of the packets $k+1$ and k computed from the past history up to the consecutive packets k and $k-1$ we have off-line calculated the error based on the real difference in the packet spacing at the receiver compared to that at the sender of the packets $k+1$ and k . As an estimation algorithm we have used the generic one defined in RTP. We used weighting factors $C = 1/16$ and $C = 1/20$ from which $C = 1/16$ is the one used in the standard. The error percentiles of the estimates using these four factors are depicted in Figure 6-23 for the measurement in the IP switching network with mixed load in the network. There are now big differences between the four factors.

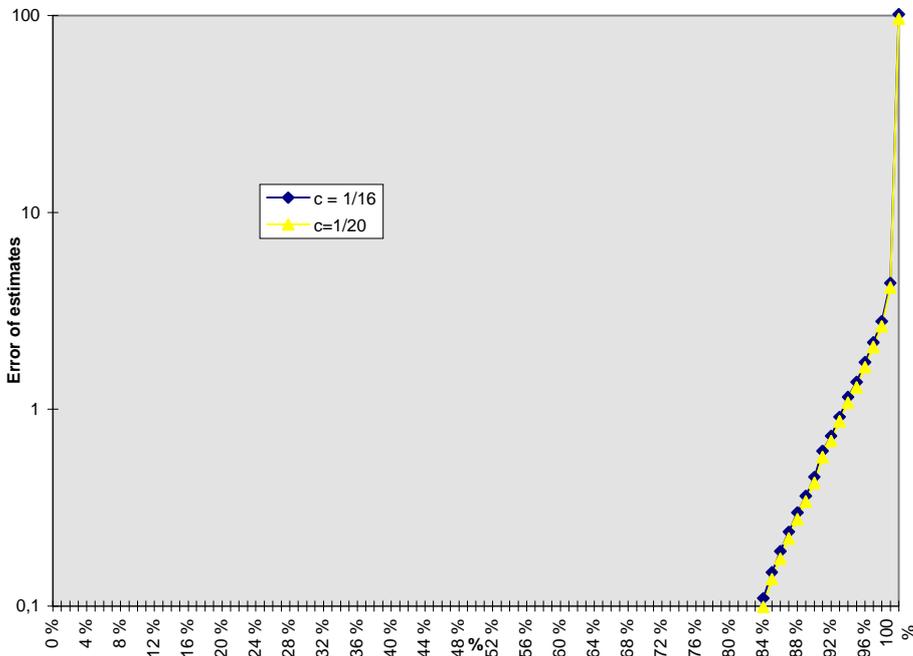


Figure 6-23: Percentiles of errors in the estimates of J using weighting factors $C = 1/16$ and $C = 1/20$ for IP-switching measurement with mixed load. Note the logarithmic scale on y-axis.

We also present the percentiles of the errors of the interarrival jitter estimates computed for the measurement in the IP switched network with two kilent-server 4:s causing background load to the network. As we presented earlier the combined load during this measurement was bursty in nature, although the sources were not bursty, and thus it presents a good starting point for us. The percentiles of the errors computed using the generic estimation algorithm and four different factors are depicted in Figure 6-24. Here we see that the 90th percentile of the errors for all the factors is around 0.1 ms.

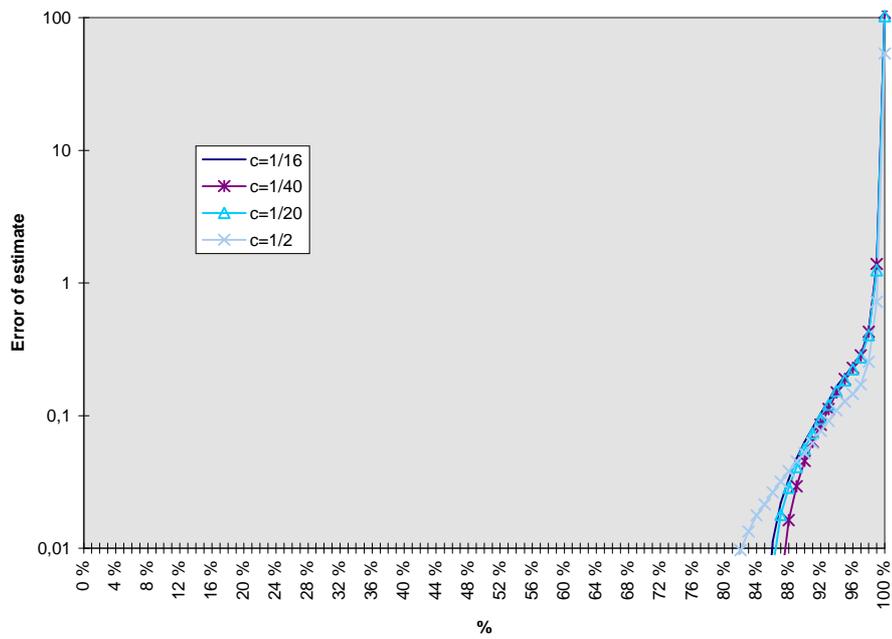


Figure 6-24: Percentiles of errors in the estimates of J using weighting factors $C=1/16$ and $C=1/20$ for IP-switching measurement with two sending kilent-server 4:s. Note the logarithmic scale on y-axis

The choice of the weighting factor affects on how quickly the estimator reacts to sudden changes in the interarrival jitter. If the weighting factor is very small, the new value is de-emphasized and the past data is emphasized. In this case, the estimator then does not react to sudden changes very quickly. If the factor is too large the estimator react to changes very well, to a point where it becomes unstable, and does not keep a steady state, i.e. it is noisy. All in all, the weighting factor in the simple generic algorithm should allow a good balance between old and new values.

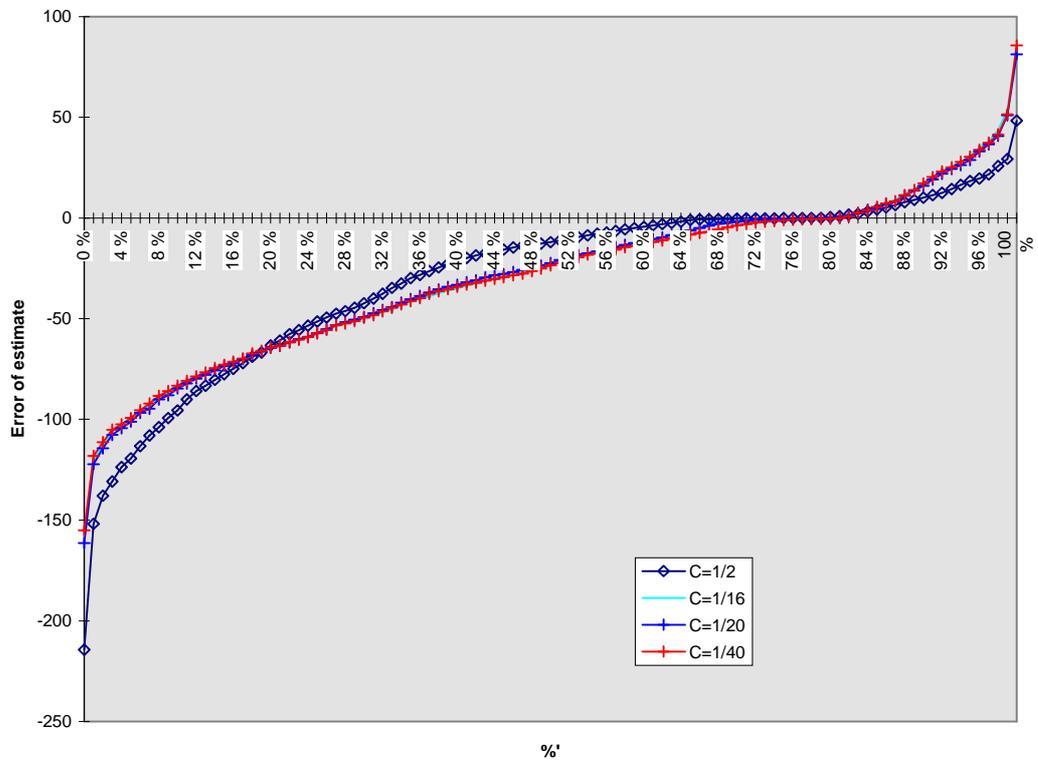


Figure 6-25: Percentiles of errors in the estimate of J using weighting factor $s C = 1/16, C = 1/20, C = 1/2$ and $C = 1/40$ calculated for measurement in Ethernet with large packet load.

In Figure 6-25 are depicted the percentiles of the errors using the same four coefficients as previously. The errors have been computed for the data of the Ethernet measurement with large packet load in the LAN. Here we can also see the effect of the spikes, the sudden changes in D . The fast responding

Measurements and Results

weighting factor of $\frac{1}{2}$ has time to react to the rise in D , but in our measurements this was not wanted. A spike was usually followed by a number of negative to zero D , because of the compression phenomenon the packets accumulate in the senders output buffer. Therefore the large factors have higher negative error values meaning that the estimate of the interarrival jitter is too large.

Although the maximum and minimum errors are a poor basis for optimization and making conclusions they do illustrate the behavior of the estimators. Summaries from a collection of measurements are presented in Table 6-12: A summary of the maximum errors of the estimates. The error is the difference between the actual D and the estimated interarrival jitter. All values are in milliseconds. and Table 6-13. We see that the maximum negative errors were consistently for the factors in the IP switched case in the order, from worst to best: $\frac{1}{2}$, $\frac{1}{16}$, $\frac{1}{20}$, $\frac{1}{40}$, and the positive errors were in the order, from worst to best: $\frac{1}{40}$, $\frac{1}{16}$, $\frac{1}{20}$ and $\frac{1}{2}$.

Are review of the estimator illustrate, that there is clearly room for more novel adaptive playout delay adjustment algorithms, and better estimation of the interarrival jitter, such as those presented in [58]. Usually within voice client programs a smoothing factor is used to multiply the estimate of interarrival jitter to get the variable delay estimate. The factor can sometimes be changed by the user. Typical values are two to four.

Table 6-12: A summary of the maximum errors of the estimates. The error is the difference between the actual D and the estimated interarrival jitter. All values are in milliseconds.

Weighting factor	FW ping (ms)	SW ping (ms)	FW 2 processes (ms)	SW 2 processes(ms)	SW mixed load (ms)
C= $\frac{1}{2}$	37.0	57.9	49.6	53.5	50.6
C=1/16	73.1	130.6	105.5	108.1	101.4
C=1/20	69.5	124.9	99.0	102.7	96.3
C=1/40	73.3	133.0	104.6	108.2	101.0

Table 6-13: A summary of the negative maximum errors of the estimates. The error is the difference between the actual D and the estimated interarrival jitter. All values are in milliseconds

Weighting factor	FW ping (ms)	SW ping (ms)	FW 2 processes (ms)	SW 2 processes(ms)	SW mixed load (ms)
C= $\frac{1}{2}$	-47.0	-67.7	-56.5	-55.8	-52.6
C=1/16	-27.4	-34.5	-19.8	-28.8	-29.5
C=1/20	-26.9	-32.7	-28.6	-27.8	-28.4
C=1/40	-23.7	-27.2	-24.4	-23.6	-24.7

7. ANALYSIS

In this analysis chapter we discuss the measurements and the results in more detail. We will compare the performance of traditional store and forward to the performance of IP switching. In section 7.2 we will discuss the measurement methods in more detail. The chapter ends with a summary.

7.1. Comparison of IP forwarding and IP switching

In this section we will first present the packet spacing difference frequencies or per packet pair arrival time differences we have calculated for the IP switching and forwarding measurements and the discuss them. After this follows a presentation and discussion of the packet spacing percentiles for the same measurements.

7.1.1. *D* frequencies

To get a better understanding of the arrival process we decided to calculate frequencies for arrival time differences *D* of all packet pairs. The distribution for different intervals of *D* are depicted for each measurement in Figures 7-1 to 7-11.

In Figure 7-1 and Figure 7-2 we have the histograms of the frequencies of *D* of a voice stream for IP forwarding and IP switching measurements with no background traffic. The histograms are tightly distributed between -0.15 and 0.15ms. The distribution of *D* for the IP switching measurement is slightly narrower. In the figures the windows size is from -1ms to 1ms. The frequencies have been calculated for intervals of 0.001ms.

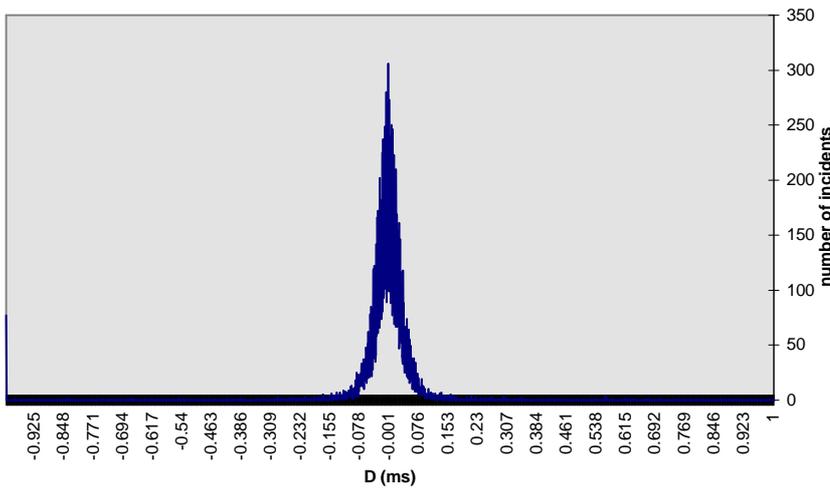


Figure 7-1: *D* frequencies of IP forwarding with no load.

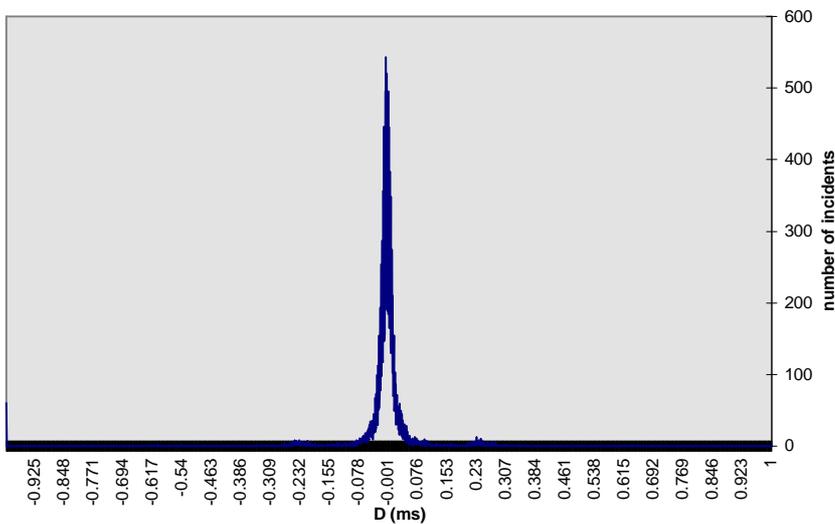


Figure 7-2: *D* frequencies of IP switching with no load.

Analysis

When we start applying load to the network the distribution of the frequencies starts widening. The window size in Figure 7-3 and in Figure 7-4 is -1ms to 1ms. The intervals for which the histogram has been calculated for is 0.001ms.

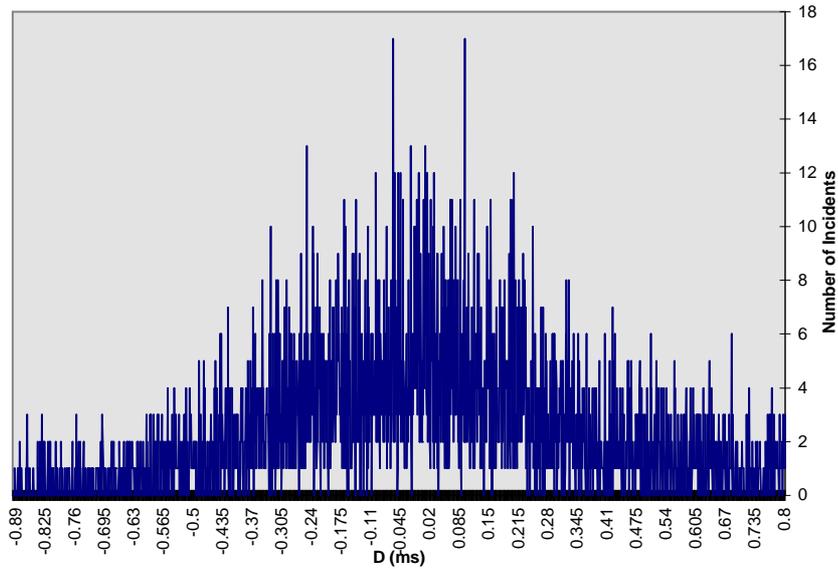


Figure 7-3: D frequencies of IP forwarding with flood ping load.

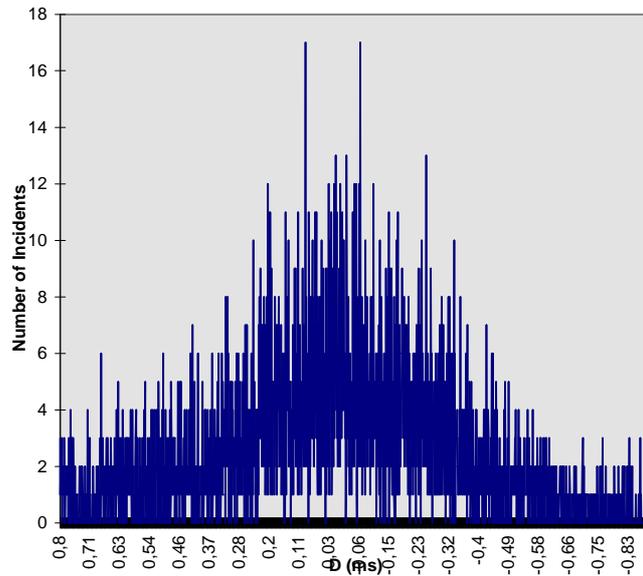


Figure 7-4: D frequencies of IP switching with flood ping load.

In Figure 7-5 to Figure 7-6 are the histograms of D calculated for measurements in IP forwarding and IP switching loaded with the Kilent-server program. If we compare results from IP forwarding and IP switching measurements, IP switching has narrower distributions of D . The reason we see any difference between the two is the small window size, combined with a fine scale. With frequencies calculated for the intervals of 1 ms, we only see spikes at 0 for both.

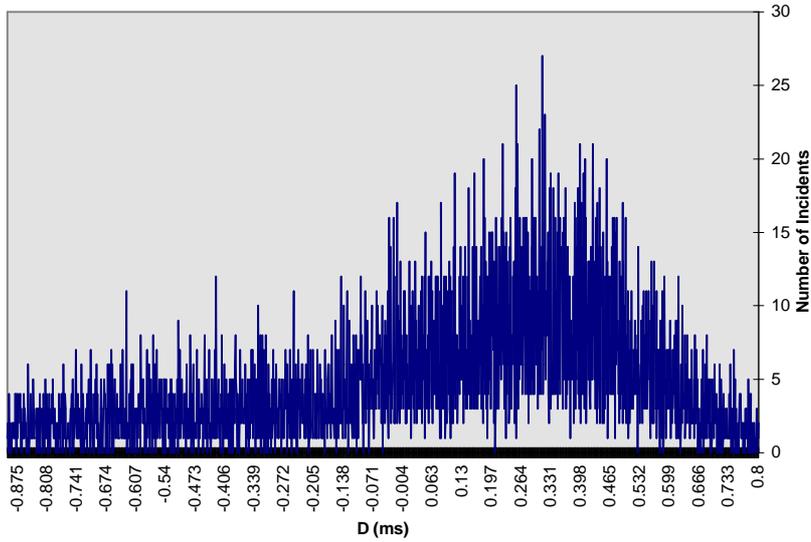


Figure 7-5: D frequencies of IP forwarding with K-S 2 processes.

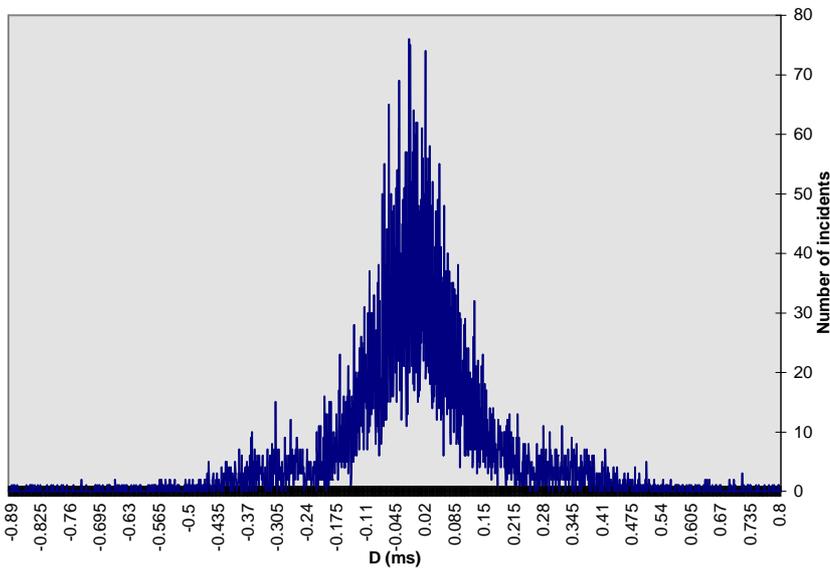


Figure 7-6: D frequencies of IP switching with K-S 2 processes.

The last histogram in Figure 7-7 was calculated for the IP switching measurement with mixed load. We see a high peak at 0 ms and the spikes at approximately 0.28 ms, -0.3 ms, 0.5 ms and -0.52 ms. The peaks are approximately 0.2 ms apart.

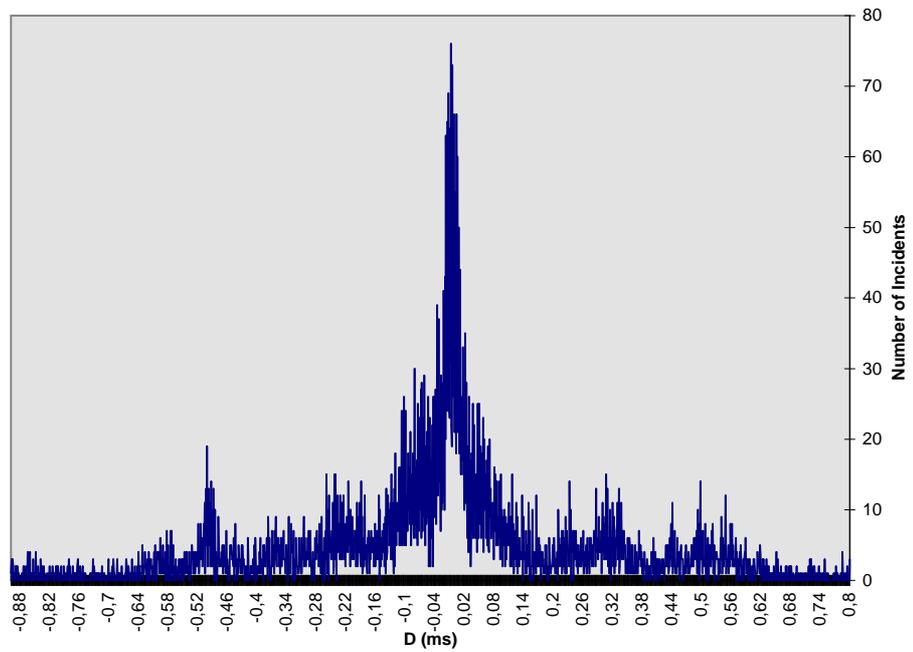


Figure 7-7: D frequencies of IP switching with mixed load.

7.1.2. D Percentiles

In order to roughly estimate the amount of playout delay adaptation that would have been needed to counter the variable component of the delay in our measurements we needed some figures on the percentiles of the packet interarrival differences. In other words what we computed was the percentage of $D < n$, where D are the packet spacing differences and n the percentage.

The percentiles of D for the non-loaded networks are depicted in Figure 7-8 and Figure 7-9. What we see is essentially a line in 0 ms going from 1% up to 99%. The network performance was very satisfactory.

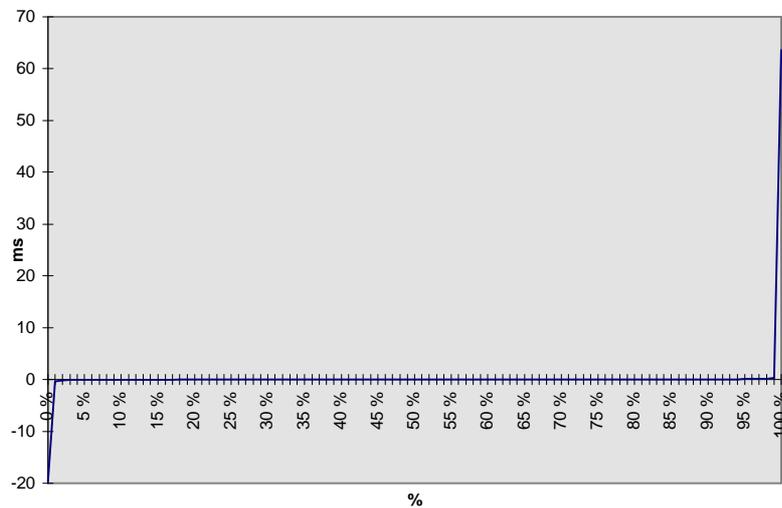


Figure 7-8: D percentiles of IP forwarding with no load ($\%D < n$ ms).

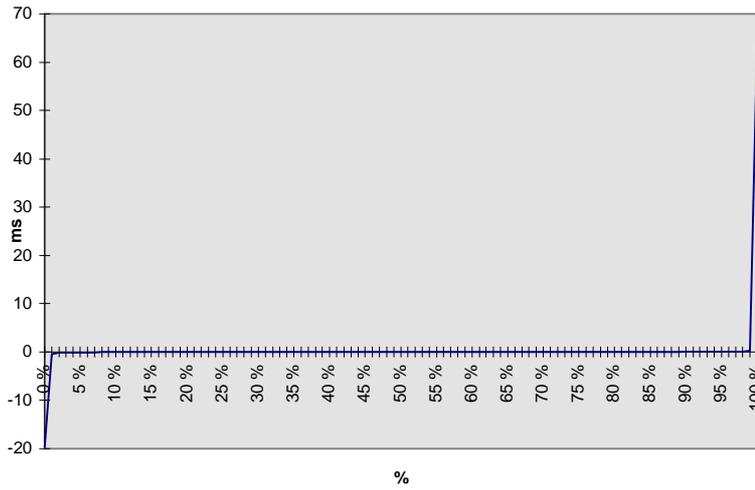


Figure 7-9: D percentiles of IP switching with no load ($\%D <n$ ms).

The percentiles of D computed from the data of the measurements where the networks were loaded with ping are in Figure 7-10 and Figure 7-11. We can see that the area of 0 ms is from 20% to 64% for IP forwarding and 20% to 81% for IP switching. For both the 90th percentile is less than 5 ms.

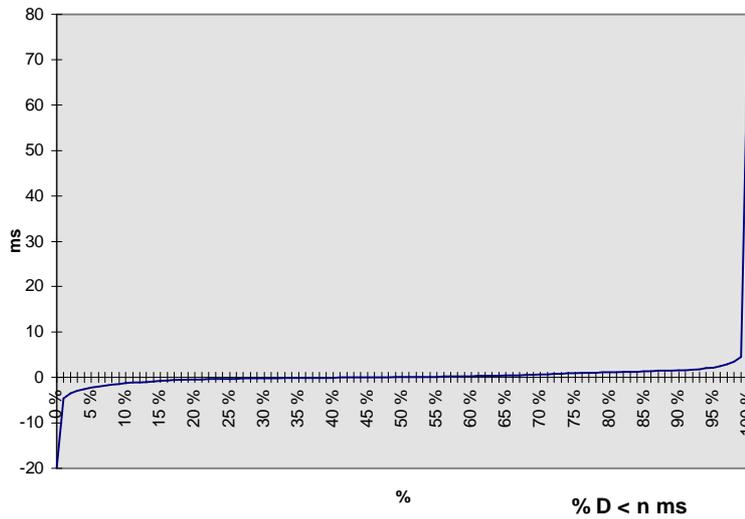


Figure 7-10: D percentiles of IP forwarding with flood ping load.

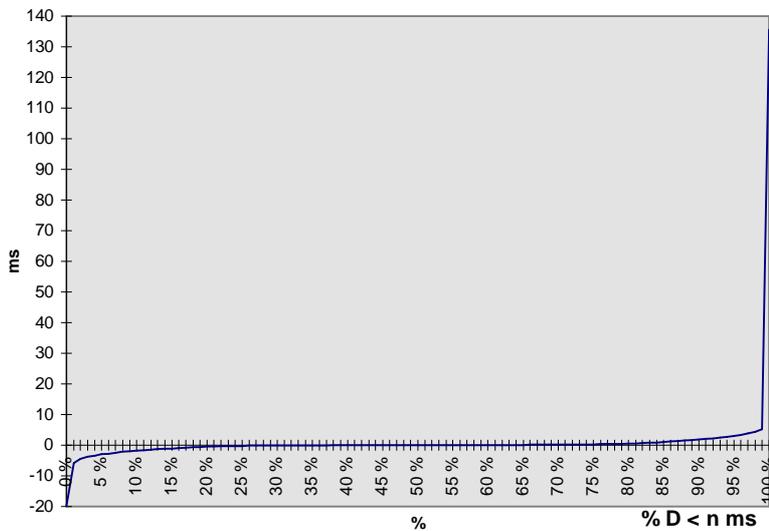


Figure 7-11: D percentiles of IP switching with flood ping load.

Analysis

In Figure 7-12 and Figure 7-13 are depicted the percentiles of D of the voice stream for measurements in the IP forwarding and IP switching networks background loaded with the kilent-server perl script. The performance is again very consistent, with the 99th percentile of D still around 0 ms for both networks.

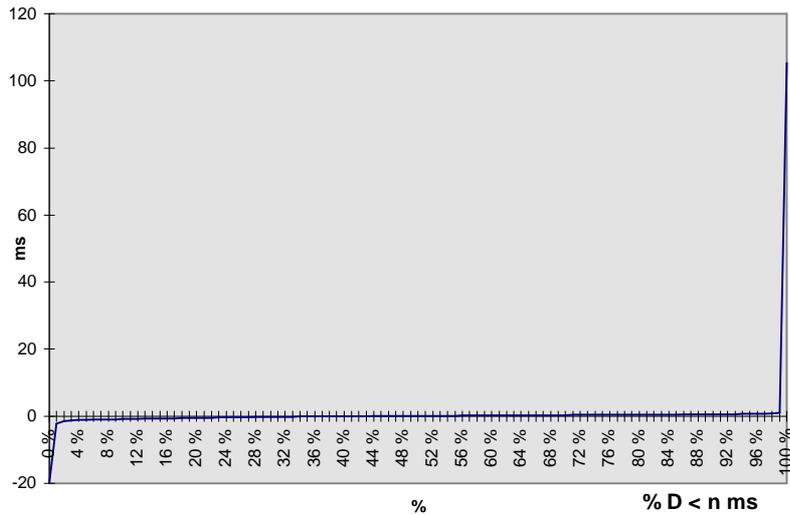


Figure 7-12: D percentiles of IP forwarding with K-S 2 processes.

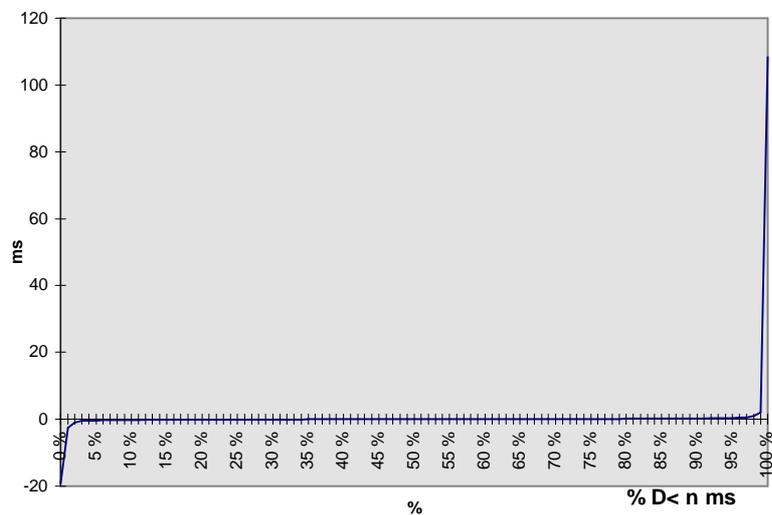


Figure 7-13: D percentiles of IP switching with K-S 2 processes.

The worst case we were able to achieve, the measurement with the network loaded with both flood ping and kilent-server did not prove to be all that bad performance wise. The 85th percentile of D is around 0 ms and the 99th percentile is around 10 ms.

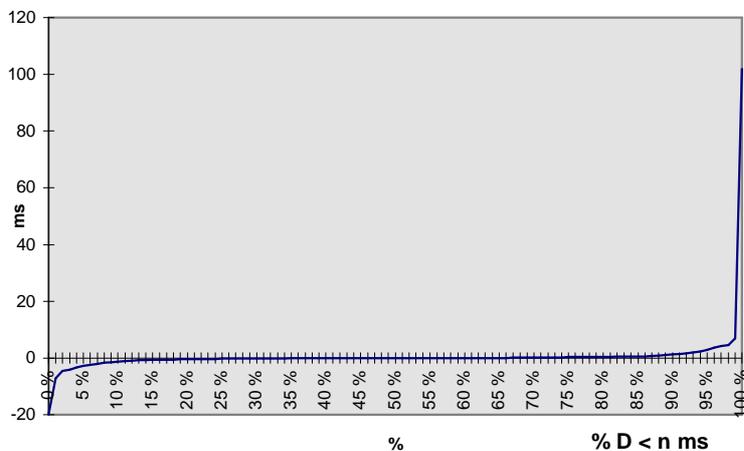


Figure 7-14: D percentiles of IP switching with mixed load.

7.2. Measurement methods

Our measurement methods have some good and bad points. Here we will discuss them and also present some alternative approaches to measuring the same parameters we have been interested in, namely delay and delay variance of the voice stream.

The actual end-to-end delay is difficult to measure. It should take into account the processing and coding delay in the workstation. In [59] end-to-end delay measurement using cross correlation of audio signals is suggested. A prerecorded stereo signal is fed into the input of the coder, sent through the network to the receiver, which can then through cross-correlation of signals approximate the delay. We did not have the possibility of implementing such a scheme, so we instead used the direct approach with specialized measurement equipment. Our approach has several difficulties. Both the sender and the receiver have to be in the same room, and not far from each other. Measurement over the Internet over continents could only be done by resending back the audio packets at some distant destination. This would give round-trip delay, not the one-way delay. One-way delay can be approximated to half of the round-trip delay, but often the delays are not symmetrical. A more serious handicap of our measurement method is, that it needs extra equipment, and extra effort. It would make sense to implement the signal correlation end-to-end delay measurement in software and integrate it in the terminal programs.

For delay variance measurement we wanted to use the same methodology as the actual VoIP applications use. Measuring the packet spacing differences for consecutive packet pairs is not the same as measuring the actual network caused delay and then calculating the delay variance from that. The latter methodology can not be used by actual applications, because it requires accurate synchronization of clocks both at the receiver and the sender, and it also requires that the clock do not wonder. The clock synchronization problem can be avoided by sending probe packets, and calculating one way delay from round-trip delay. This generates unwanted overhead to the network.

We did not use actual network traffic for loading the network, but instead decided to generate wanted traffic with special applications. Typical campus network traffic profile would be more bursty in nature than the one we used. The traffic we used in the measurements has the benefit that it is can be easily reproduced. The packet length distribution in our measurement represents the typical case with a number of maximum length packets and also a large number of short packets.

We generated the background traffic the IP switch gateways. The traffic generation and forwarding into the network is a different process than only routing and forwarding. Our results do not represent accurately the case where we would have LANs on both ends of the network loading the backbone. For example the throughput figures should not be taken as such. Our measurements give the difference between IP switching and IP forwarding used in the core of the network, in the actual IP switches. We only had two switches in the core, and therefore the expected differences were not big. Also the amount of traffic was not exactly the same in the IP switched and forwarded measurements. The throughput increased, when IP switching was used, and both flood ping and kilent-server started sending more traffic.

7.3. Summary

In this chapter we analyzed in further detail the results from our measurements of the packet spacings (D) of a voice stream over IP forwarding and IP switching. First in section 7.1.1 we presented the histograms of D for both environments and all measurements. From the histograms we could see that voice over IP switching has a narrower distribution of D than voice over IP forwarding. After the histograms, in section 7.1.2 we presented the percentiles of D for our measurements. Finally in section 7.2 we discussed our measurement methods and presented some alternative ways of measurement.

CONCLUSIONS AND FUTURE WORK

This thesis focused on the delay and delay variance of IP Voice. The end-to-end delay consisting of network and terminal delays of IP Voice over a campus network and a LAN was measured. Also, the network caused delay variance of the IP Voice packets was analyzed from measurements in Ethernet, a routed network and a network of IP switches using IP flow cut-through switching. In addition the performance of the basic interarrival jitter estimation algorithm defined in RFC 1877 was studied with different weighting factors.

We measured the total end-to-end delay from the sender input to the receiver output of IP voice over Ethernet and over an IP switching network. We also measured the terminal delay, the terminal hardware delay and the network delay separately. The total end-to-end delay was approximately equal to the terminal delay. The network delay was negligible as was the audio hardware delay. The end-to-end delay of IP Voice over a LAN and a campus network are mainly caused in the terminal between digitization of the audio and transmission of the IP-packets, i.e. the bus, buffering, coding, RTP-, UDP-, and IP header processing.

We measured packet spacing differences at the receiver compared to the fixed difference at the sender of the voice packets over Ethernet, IP forwarding and IP switching. When the networks were free of other traffic the spacing difference was 0 for all three networks for 99% of the packets. We applied controlled load to the network and even over a highly loaded network the packet spacing differences stayed under 10 ms for 99% of the packets. The IP switching network was slightly better than a IP forwarding network. In our measurements the edge-router performance was a predominant factor. The workload of the edge-router, the IP switch gateway does not decrease significantly, when cut-through switching is used in the core network.

The packet spacing frequency histograms were also presented. The histograms showed the same thing as the percentiles: the packet spacing difference of zero dominates, and that when we apply more load to the network the frequency histogram widens to the point where we have white noise like frequency distribution with a peak in zero, as was the case for Ethernet loaded with a large packets. If we take a small enough intervals for the histogram, 0.001 ms, from -1ms to 1 ms we see that there is in fact some difference between IP forwarding and IP switching. The packet spacing differences and thus the interarrival jitter accumulates when nodes are added. We only had four nodes in our network so increasing the number of nodes would result in larger differences between IP switching and IP forwarding. We also derived the basic algorithm for estimation of the delay variance of the packets. We calculated delay variance estimates for all our measurement data using four different coefficients. In our measurements the network performance is so consistent that a fixed playout delay of 0-5 ms would suffice, and in general give better performance than added variable delay based on delay variance estimation. The number value of the weighting factor used in the estimation algorithm of RTCP, $1/16$ was not optimal in our measurements. For example with the factor of $1/20$ the packet loss would have been smaller, and the playout delay shorter. We did not try to find the optimal coefficient, because our view was that the optimal coefficient varies with the network load and the network, and so an adaptive algorithm with an adaptive memory factor would be better. A very good study on this is in [58]. As a final conclusion voice over IP seems to be evolving rapidly, and is already now a viable option for voice communication. The end-to-end delay is at best non noticeable (around 100 ms) and the delay variance is very small in LANs and campus networks.

Future work

This work concentrated on objective IP Voice quality parameters. There has been some work on subjective quality of packetized voice, but linking subjective end-to-end quality with objective parameters still requires work. For example what delay and delay variance are tolerable? ITU-T has developed Perceptual Speech Quality Measure-code, which incorporates a model of the human speech quality perception. The PSQM is intended e.g. for evaluating the performance of different speech codecs. Applying PSQM for end-to-end VoIP quality could open new areas of research.

The IP voice terminals have been said to have unsatisfactory overall sound quality.

It was noted that in our case most of the end-to-end delay was accumulated in the terminal. What exactly causes the delay? Is delay the cause of slow and non realtime coding of the speech samples requiring excessive buffering, or is the problem in the transferring of voice samples from the soundcard to memory or is it in RTP, UDP and IP header processing.

In the playout delay estimation part we only scratched the surface. We could apply modern DSP algorithms in to the estimation of the different parameters. For example adaptive weighting using estimation error as input might be interesting.

New, faster and cheaper processors are coming to the market continuously, and it would make sense to keep an eye on the developments and possibly measure and evaluate the new PC architectures as the emerge.

APPENDIX A: SOME MORE FIGURES

Here are some additional figures for those interested. The captions should be self explanatory, and referring to the appropriate sections in the text can also help.

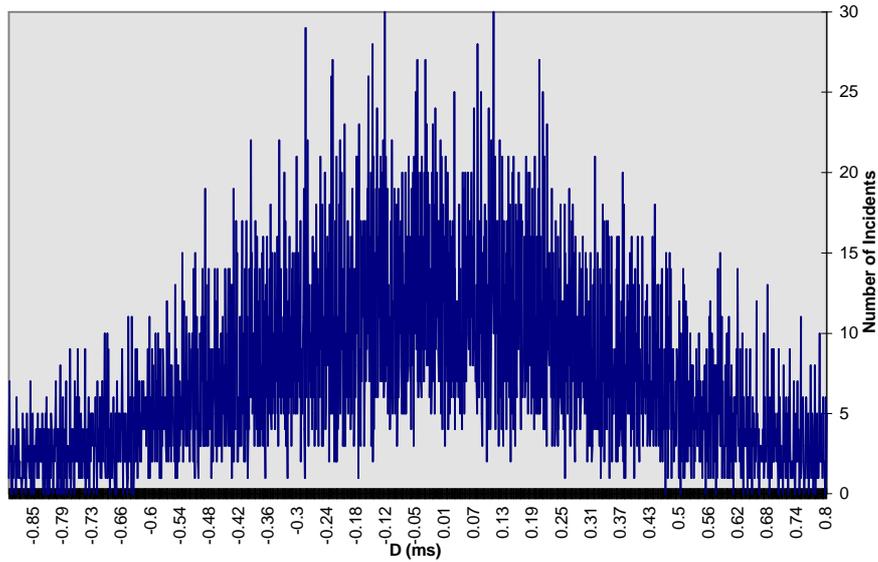


Figure 7-15: D frequencies of IP forwarding with K-S 4 processes.

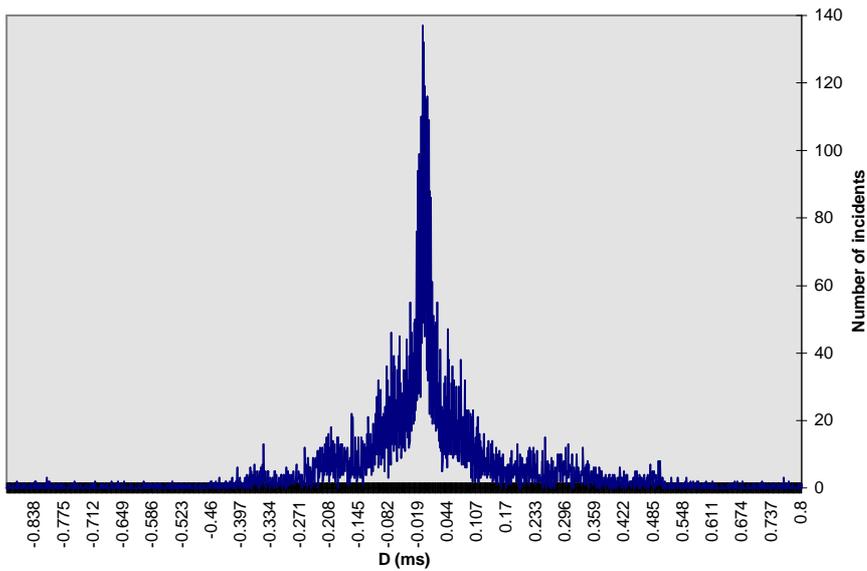


Figure 7-16: D frequencies of IP switching with K-S 4 processes.

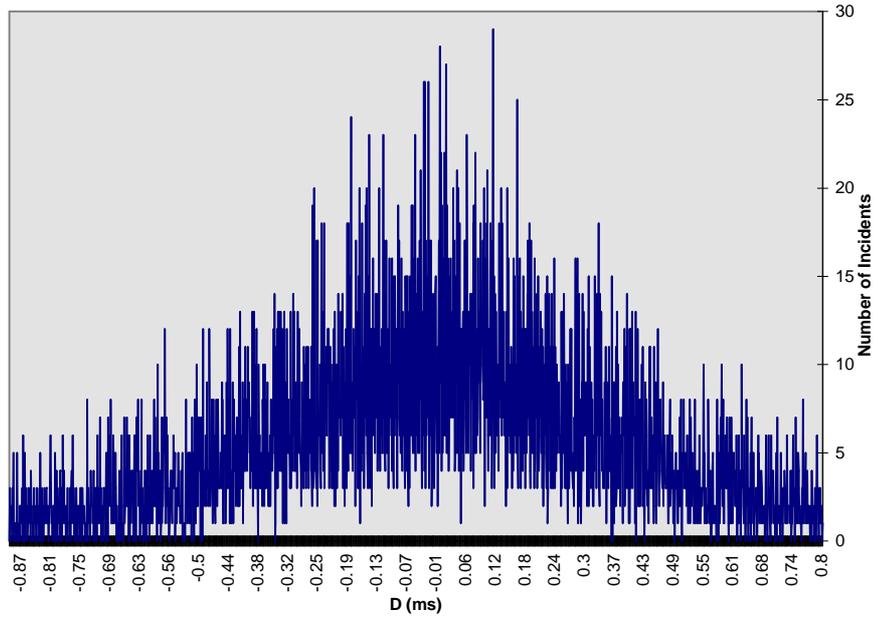


Figure 7-17: D frequencies of IP forwarding with K-S 8 processes.

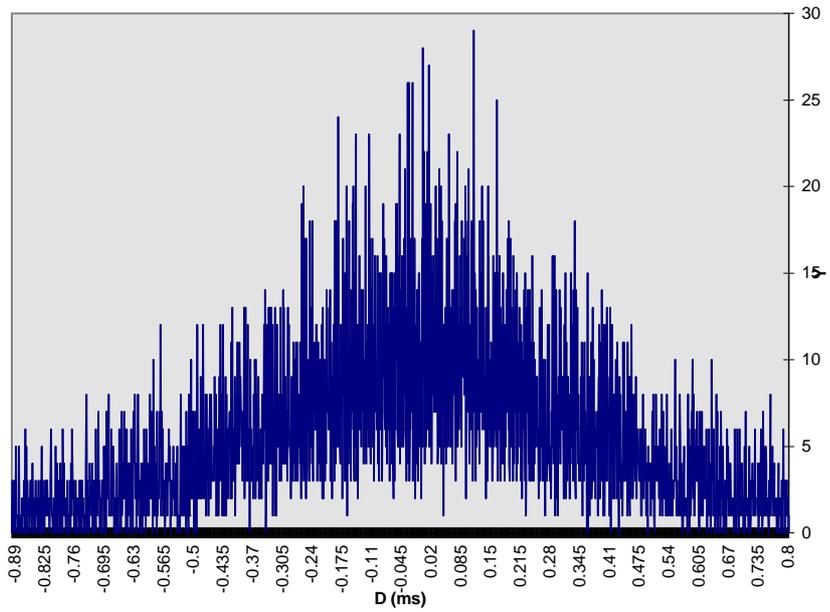


Figure 7-18: D frequencies of IP switching with K-S 8 processes.

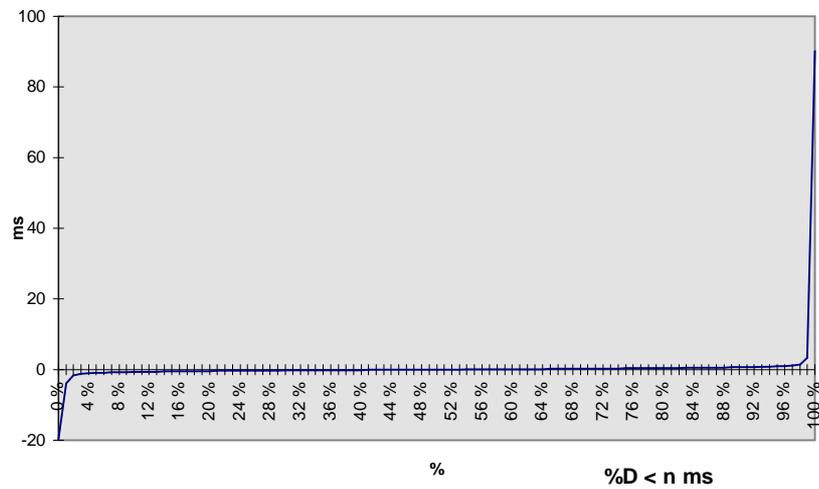


Figure 7-19: D percentiles of IP forwarding with K-S 4 processes.

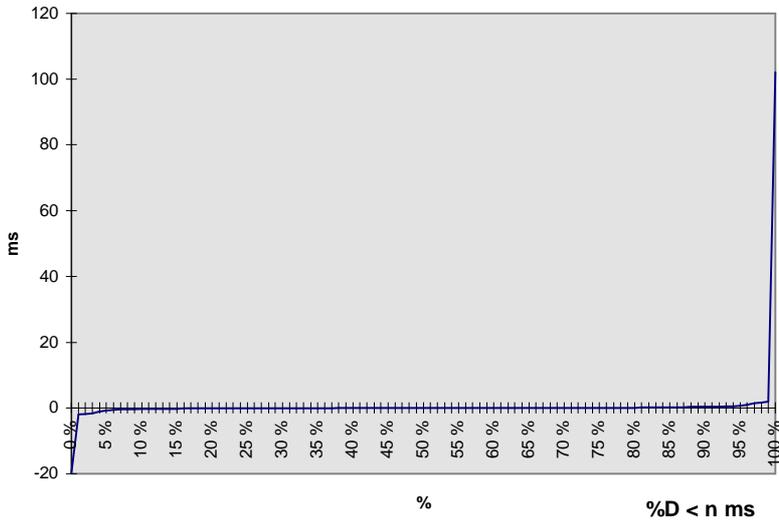


Figure 7-20: D percentiles of IP switching with K-S 4 processes.

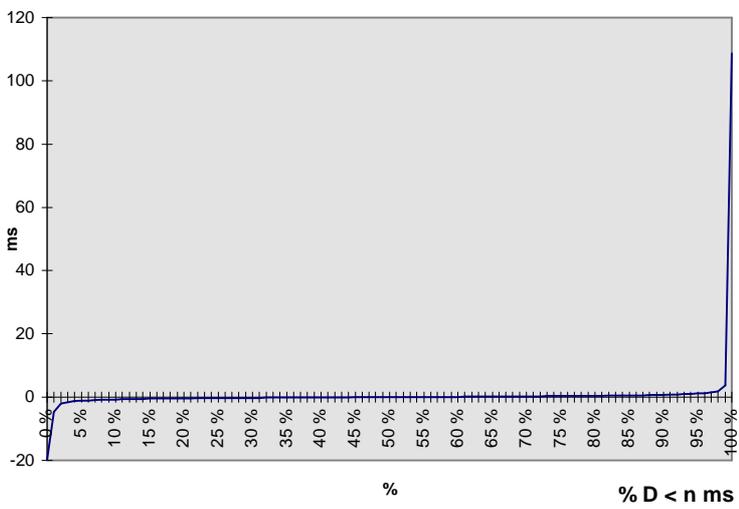


Figure 7-21: D percentiles of IP forwarding with K-S 8 processes.

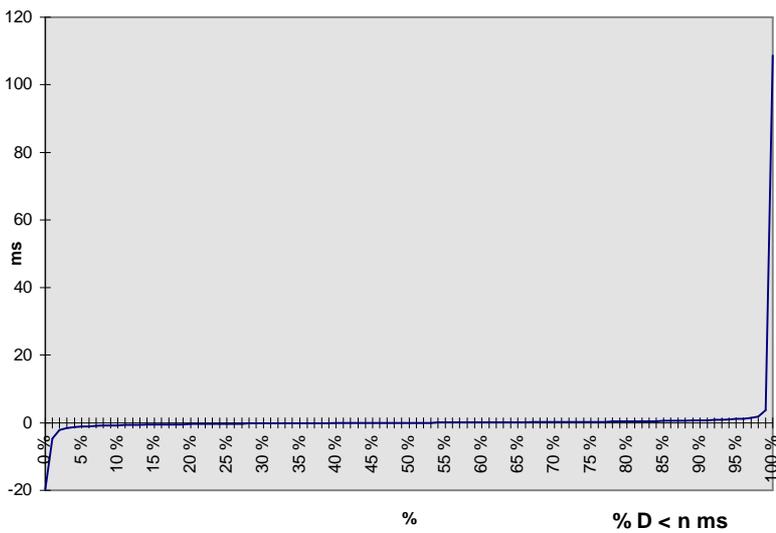


Figure 7-22: D percentiles of IP switching with K-S 8 processes.

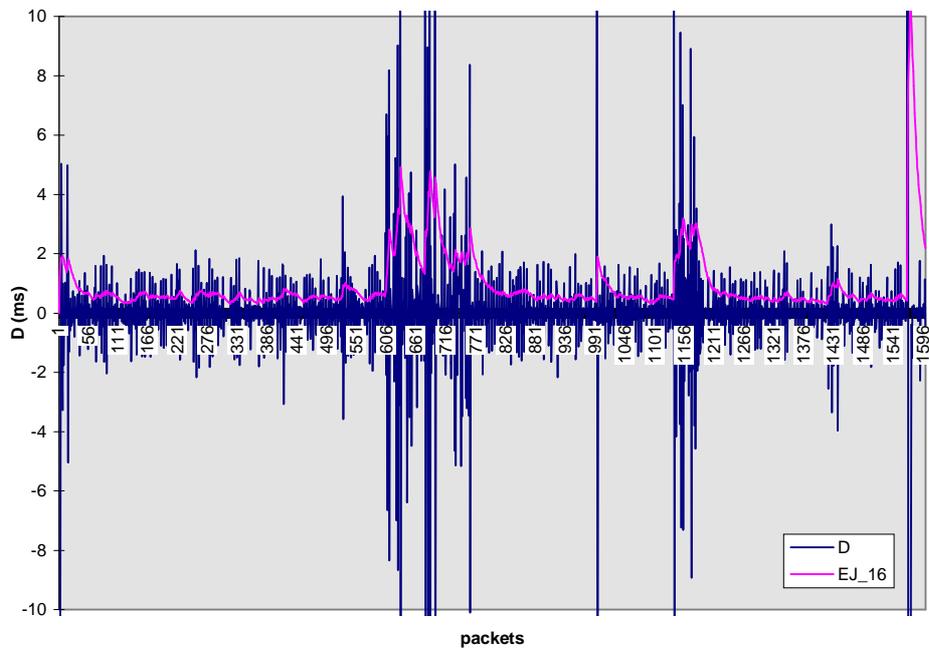


Figure 7-23: Packet spacing differences of IP forwarding with K-S 8 processes

APPENDIX B: IP SWITCHING WITH PRIORITIES

In IP switches it is also possible to assign priorities to flows. When priorities are in use the switch uses priority queuing between classes and FIFO inside a class. The priorities are: low, normal and high. We experimented with the priorities by assigning all UDP traffic to be switched and having high priority. All other traffic that was identified to be long lived (more than five packets) was switched. The result was that around 100% of traffic was switched and all voice traffic was given high priority. The percentiles and frequencies of D of the measurements are depicted in 7-24 to 7-27. The loads used were two sending processes of kilent-server 4, and mixed load of kilent-server 4 and flood ping. If we compare these results with the results of the measurements with all traffic switched but no priorities given, there are some differences. The histogram is not as dispersed and the percentile of 0 D is somewhat higher. The priority queuing seems to work as it is supposed to: voice does get priority over the background traffic.

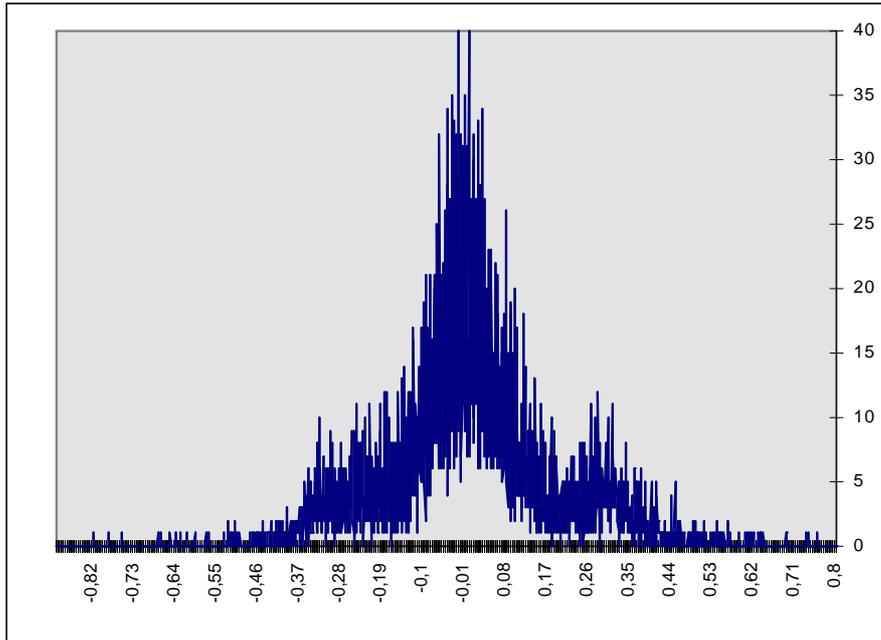


Figure 7-24: Frequencies of D of the voice stream. Voice traffic has priority. Background load is generated with two kilent-servers.

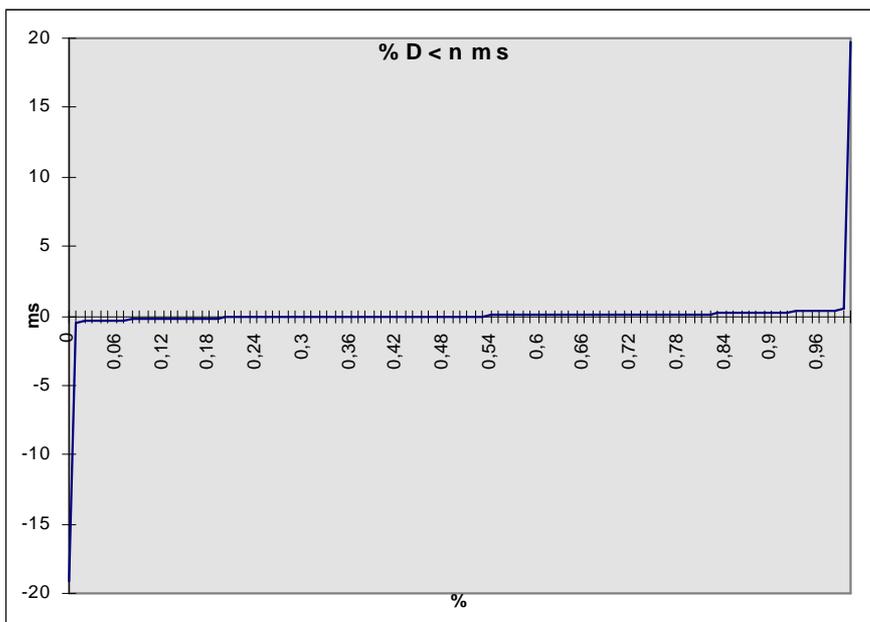


Figure 7-25: Percentiles of D of the voice stream. Voice is give higher priority than other traffic. Background load is generated with two sending processes of kilent-server-script.

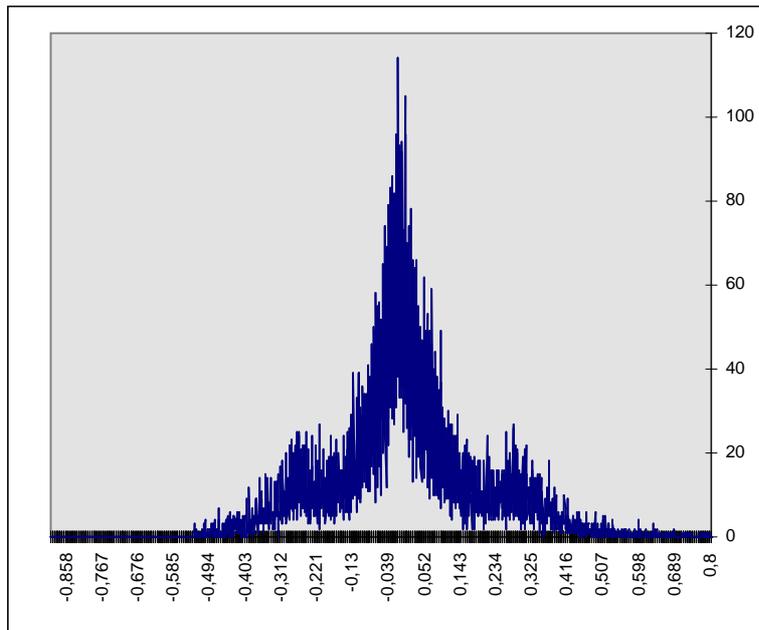


Figure 7-26: Frequencies of D . Voice has been given higher priority, and the background load is generated both with kilent-server and flood ping.

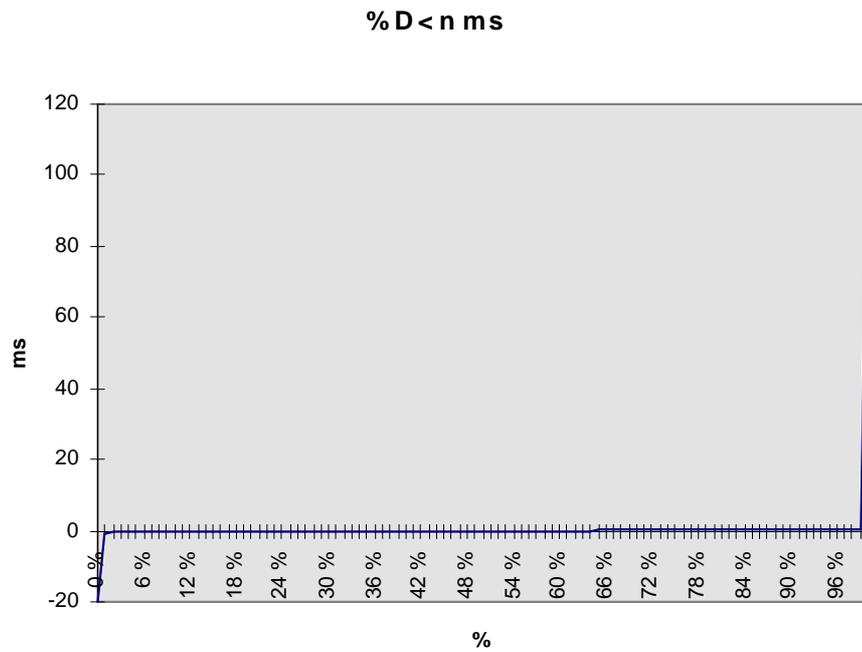


Figure 7-27: Percentiles of D . Voice has been given higher priority, and the background load is generated both with kilent-server and flood ping.

REFERENCES

- [1] Gold, B., "Digital Speech Networks", IEEE Proceedings, vol. 65, no. 12, Dec. 1977
- [2] Postel, J., "Internet Protocol, DARPA Internet Program Protocol Specification", Network Working Group, Request Request for Comments:791, Sep. 1981.
- [3] Huitema, C., "Routing in the Internet", Prentice Hall, New Jersey, USA, 1995.
- [4] Deering, S., Hinden, R., Internet Protocol, Version 6 (IPv6) Specification , RFC 1883, Dec. 1995.
- [5] Deering, S., Hinden, R., Internet Protocol, Version 6 (IPv6) Specification , Internet Draft, draft-ietf-ipngwg-ipng-spec-v2-00.txt , July 1997.
- [6] Schulzrinne, H., "Issues in Designing a Transport protocol for Audio and Video Conferences and other Multiparticipant Realtime Applications", Audio-Video Transport Working Group Internet-draft , Internet Engineering Taskforce, October 1993.
- [7] Shulzrinne, H., Casner, S., Frederick, R., Jacobsen, V., "A Transport Protocol for Realtime Applications", Network Working Group Request for comments:1889, Internet Engineering Task Force , Jan. 1996.
- [8] ITU-T, "H.225.0 Line Transmission of Non-Telephone Signals",pp.9-118, May 1996.
- [9] Thom, G., "H.323: The Multimedia Communication Standard fo Local Area Networks", IEEE Communications, Dec. 1996.
- [10] Parulkar, G., Schmidt, D.C., Turner, J.S., " IP/ATM: A strategy for integrating IP with ATM", Proc. SIGCOM, Cambridge, MA, p.9, Sep. 1995.
- [11] Esaki, H, Ohta, M., Nagami, K., "High Speed Datagram Delivery over Internet using ATM", IEICE TRANS. Communications, Vol. E78-B, No.8, Aug 1995.
- [12] Rekhter, Y., Davie, B., Kats, D., Rosen, E., Swallow, G., "Cisco Systems' Tag Switching Architecture Overview", Request for comments:2105, Network Working Group, Cisco Systems, Inc., 1997.
- [13] Viswanathan, A., Feldman, N., Boivie, R., Woundy, R., "ARIS: Aggregate Route-Based IP Switching" Internet draft, Network Working Group, Internet Engineering Task Force, Mar.1997, draft-viswanathan-aris-overview-00.txt.
- [14] Rosen, E., Viswanathan, A., Callon, R., "A Proposed Architecture for MPLS", Internet draft, Network Working Group, Internet Engineering Task Force, Aug. 1997, draft-ietf-mpls-arch-00.txt.
- [15] ATM Forum "Multiprotocol over ATM", 1997.
- [16] Newman, P., Minshall, G., Lyon, T., Huston, L., "IP Switching and Gigabit Routers", IEEE Communications, pp.64-69, Jan. 1997.
- [17] Ilvesmäki, M., Kilkki, K., Luoma, M., "Packets or ports - the decisions of IP switching", Proc. Voice, Video and Data Communications, SPIE'97, Dallas, TX, Nov. 1997.
- [18] Ilvesmäki, M., "The use of ATM-technology in switching of internet-traffic", Masters's Thesis, Lab. Of Telecommunications Technology, Helsinki University of Technology, Nov. 1996.
- [19] Rabiner, L., "Applications of Voice Processing to Telecommunications", Proceedings of the IEEE, Vol. 82, No.2, Feb. 1994.
- [20] AT&T, "G.728 decoder modifications for frame erasure concealment", document no. AH-5-10, ITU-T Rapporteur's meeting, Tokyo, Japan, Mar. 1994.
- [21] Cox, R., "Three New Speech Coders from the ITU Cover a Range of Applications", IEEE Communications, Vol.35, No. 9 , pp. 40-47, Sep. 1997.
- [22] Perkins, M., Evans, K., Pascal, D., Thorpe, L., "Characterizing the Subjective Performance of the ITU-T 8 kb/s Speech Coding Algorithm - ITU-T G.729", IEEE Communications, Vol.35, No. 9 , pp. 74-81, Sep. 1997.
- [23] ITU-T, "Recommendation E.800: Quality of service and dependability vocabulary", 1988.
- [24] ISO/IEC, "DIS 13236: Information Technology Quality of Service", 1996.
- [25] Clark, Shenker, Zhang, "Supporting Realtime Applications in an Integrated Services Packet Network: Architecture and Mechanism", Laboratory of Computer Science, Massachusetts Institute of Technology, 1996, <http://ana-www.lcs.mit.edu/anaweb/pdf-papers/CSZ.pdf>

-
- [26] Coviello, G., "Comparative Discussion of Circuit -vs. Packet-Switched Voice", IEEE Transactions on Communications, vol.COM-27, no.8, pp.1153-1159, Aug.1979
- [27] IMTC Voice over IP Forum, "Service Interoperability Implementation Agreement (draft)", Mar. 1997, VoIP98-008.doc.
- [28] Kotikalapudi Sriram, Whitt, W., "Characterizing Superposition Arrival Process in Packet Multiplexers for Voice and Data", IEEE Journal on Selected Areas in Communications, vol.SAC-4, no.6, pp. 833-845, Sept.1986.
- [29] Heffes, H., Lucantoni, D., "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance", IEEE Journal on Selected Areas in Communications, vol.SAC-4, no.6, pp. 856-867, Sept.1986.
- [30] Daigle, J., Langford, J., "Models for Analysis of Packet Voice Communications Systems", IEEE Journal on Selected Areas in Communications, vol.SAC-4, no.6, pp. 847-855, Sept.1986.
- [31] Bolot, J-C, "Characterizing End-to-End Packet Delay and Loss in the Internet", Proc. ACM Sigcomm '93, San Fransisco, California, pp.289-298, Sept. 1993.
- [32] Schwartz, M., "Broadband Integrated Networks", pp.21-58, Prentice Hall, NJ, USA, 1996.
- [33] Mills, D., L., "Internet Delay Experiments", Request for comments 889, Network Working Group, Internet Engineering Task Force, Dec. 1983.
- [34] Steimetz, R., "Human Perception of Jitter and Media Synchronization", IEEE Journal on Selected Areas in communications, vol.14, no.1, pp. 61-72, Jan. 1996.
- [35] Boggs, D., Mogul, J., Kent, C., "Measured Capacity of and Ethernet", Research Report 88/4, Western Research Laboratory, Palo Alto, CA, Sep. 1988.
- [36] Gruber, J., Strawczynski, L., "Subjective Effects of Variable Delay and Speech Clipping in Dynamically Managed Voice Systems", IEEE Transactions on Communications, vol.COM-33,no.8, pp.801-809, Aug.1985.
- [37] Jayant, N.,Christensen, S., "Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation", IEEE Transactions on Communications, Vol. COM-29, No.2, Feb. 1981.
- [38] Sherif, M., Crossman, A., Jeffrey, R., "Document T1A1.7/94-016r2:A technical report on speech packetization", T1 Committee, T1A1.7, WG on Specialized Signal Processing, 1994.
- [39] Goodman, J., Goodman, G.B., Dvorak, C. A., Page, H. G., "Waveform substitution techniques for recovering missing speech segments in packet voice communications", IEEE Transaction on Acoustic Speech, Signal Processing, Vol. ASSP-34, No.6, pp.1440-1448, Dec.1986.
- [40] Bolot, J.-C., Vega-García, A., "The Case for FEC-Based Error Control for Packet Audio in the Internet", to be published in ACM Multimedia Systems.
- [41] Bolot, J.-C., Vega-García, A., "Control Mechanisms for Packet Audio in the Internet, Proc. IEEE Infocom '96, San Francisco, California, pp.232-239, Apr. 1996.
- [42] Bolot, J.-C., Hugues, C., "Analysis and Control of Audio Packet Loss over Packet-Switched Networks", Inria, Sophia Antinopolis, France, 1993.
- [43] Luoma, M., Ilvesmäki, M., "Simplified management of ATM traffic", Proc. SPIE'97, Voice, Video and Data Communications, Dallas, TX, Nov. 1997.
- [44] Kilkki, K., "Simple Integrated Media Access (SIMA) ", Internet-Draft, Jun. 1997, draft-kalevi-simple-media.access-01.txt.
- [45] Shenker, S., Partridge, C., Guerin, R., "Specification of Guaranteed Quality of Service", Integrated Services Working Group, Internet-Draft, Feb.1997, draft-ietf-intserv-guaranteed-svc-07.txt.
- [46] Wroclawski, J., "Specification of the Controlled-Load Network Element Service", Integrated Services Working Group, Internet-Draft, Nov..1997, draft-ietf-intserv-ctrl-load-svc-04.txt.
- [47] Crowcoft, J., Wang, Z., Smith, A., Adams, J., "A Rough Comparison of the IETF and ATM Service Models", IEEE Network, Vol.9, No.6, Nov./Dec. 1995, pp.12-16
- [48] Sanghi, D., Agrawala, A. K., Gudmundsson, Ó, Jain, B. N., "Experimental Assesment of End-to-end Behavior on Internet", Technical Report CS-TR-2909, Dept. of Computer Science, University of Maryland, 1992.
- [49] Pointek, J., Shull, F., Tesoriero, R., Agrawala, A., "NetDyn Revisited: A Replicated Study of Network Dynamics", Technical Report, Dept. of Computer Science, University of Maryland, Oct. 1996.
- [50] Borella, M., Sidhu, I., "Self-Similarity of Internet Packet Delay", Proc. IEEE, pp. 513-517, Aug. 1997.
- [51] Agrawala, A., Sanghi, D., "Network Dynamics: an Experimental Study of the Internet", Technical Report CS-TR-3696, Dept. of Computer Science, University of Maryland, 1993.

-
- [52] Maxemchuk, N. F., Lo, S., "Measurements and Interpretation of Voice Traffic on the Internet", Proc. ITC '97, Sidney, Australia, May 1997.
- [53] The NLANR homepage can be found in the url: <http://www.nlanr.net>
- [54] The MICE home page is in the url:<http://www.cs.ucl.ac.uk/mice>
- [55] Busse, I., Deffner B., Shulzrinne, H., "Dynamic QoS Control of Multimedia Applications based on RTP", Technical Report, GMD-Fokus, May 1995.
- [56] Sisalem, D., "End-to-end Quality of Service Control Using Adaptive Applications", Proc. IFIP Fifth International Workshop on Quality of Service, IFIP WG 6.1, May 1997.
- [57] Ramjee, R., Kurose, J., Towsley, D., Shulzrinne, H., "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks", Proc. IEEE Infocom '94, Montreal, Canada, Apr. 1994.
- [58] Moon, S., Kurose, J., Towsley, D., "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms", Technical Paper, Dept. of Computer Science, Univ. of Massachusetts at Amherst,
- [59] Shulzrinne, H., "Internet Telephony - Towards the Integrated Services Internet", Workshop on Internet Telephony, Utrecht, The Netherlands, Feb. 1996.