KTH Electrical Engineering

Quality aspects of Internet telephony

IAN MARSH

Doctoral Dissertation
Stockholm, Sweden 2009

TRITA-EE 2009:025
ISSN: 1653-5146
ISRN KTH/EE–09/025–SE
ISBN 978-91-7415-313-2

School of Electrical Engineering
KTH, Stockholm, Sweden

Akademisk avhandling som med tillstånd av Kungliga Tekniska Högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i telekommunikation fredagen den 5 juni 2009 vid KTH.

Tryck: Universitetsservice US AB

Swedish Institute of Computer Science, SE–164 29 Kista, SWEDEN
SICS Dissertation Series 51
ISSN-1101-1335
ISRN SICS-D–51–SE

Abstract

Internet telephony has had a tremendous impact on how people communicate. Many now maintain contact using some form of Internet telephony. Therefore the motivation for this work has been to address the quality aspects of *real-world* Internet telephony for both fixed and wireless telecommunication. The focus has been on the quality aspects of voice communication, since poor quality leads often to user dissatisfaction. The scope of the work has been broad in order to address the main factors within IP-based voice communication.

The first four chapters of this dissertation constitute the background material. The first chapter outlines where Internet telephony is deployed today. It also motivates the topics and techniques used in this research. The second chapter provides the background on Internet telephony including signalling, speech coding and voice Internetworking. The third chapter focuses solely on quality measures for packetised voice systems and finally the fourth chapter is devoted to the history of voice research.

The appendix of this dissertation constitutes the research contributions. It includes an examination of the access network, focusing on how calls are multiplexed in wired and wireless systems. Subsequently in the wireless case, we consider how to handover calls from 802.11 networks to the cellular infrastructure. We then consider the Internet backbone where most of our work is devoted to measurements specifically for Internet telephony. The applications of these measurements have been estimating telephony arrival processes, measuring call quality, and quantifying the trend in Internet telephony quality over several years. We also consider the end systems, since they are responsible for reconstructing a voice stream given loss and delay constraints. Finally we estimate voice quality using the ITU proposal PESQ and the packet loss process.

The main contribution of this work is a systematic examination of Internet telephony. We describe several methods to enable adaptable solutions for maintaining consistent voice quality. We have also found that relatively small technical changes can lead to substantial user quality improvements. A second contribution of this work is a suite of software tools designed to ascertain voice quality in IP networks. Some of these tools are in use within commercial systems today.

Acknowledgments

Two pages of acknowledgments, "Oh please".

The first line of the acknowledgment section in my 2003 licentiate thesis reads "Writing this part of the thesis is actually *enjoyable*." Now, in April 2009, for my doctoral dissertation the best I can come up with is "Writing this part of the dissertation is actually *weird*." I could *never* have imagined how much more effort there was still remaining, as well as the ups and downs that would accompany them.

Some people have been responsible for getting me close to the end and they are first and foremost Prof. Gunnar Karlsson who enrolled me, kept with me, and hopefully will see me graduate. Without him none of the eight years of PhD studies would have ever happened. Also thanks to Dr. Bengt Ahlgren, my boss and lab leader of the NETS group at SICS, again without whom I would not be at this point. Thanks to you both! Acknowledgments also to Prof. Gerald Q. "Chip" Maguire Jr. whose input and influence is present within this dissertation.

I would also like to acknowledge Janusz Launberg the business manager and Dr. Staffan Truvé the CEO at SICS. Thanks for the support over the years. I would like to thank the many other people at SICS for the creative and relaxing environment. This includes all the support staff, which seem to be sadly overlooked in many acknowledgments. The group(s) within which one works are critical, therefore the folks of NETS (formerly CNA) and the chaps at LCN deserve a special mention, some of which have become good friends, which goes to show there is more to life than just research (but not much more). To the original LCN'ers we have (almost) made it.

Some of this work has been done in collaboration with people namely Olof Hagsand, Florian Hammer, Christian Hoene, Ingemar Kaj, Moo Young Kim, and Martín Verala it was a pleasure to work with you all. The students I was responsible for during the years (chronologically) are: Zheng Sun, Anders Gunnar, Fengyi Li, Juan Carlos Martín Severiano, Viktor Yuri Diogo Nunes and Daniel Lorenzo, all of whom have been successful in their post education lives. It was a great pleasure to be involved in your education and I hereby gratefully acknowledge your contribution in my PhD dissertation.

The Swedish PhD presents an opportunity to do research. It also presents an opportunity to develop highly needed technical skills in the form of

courses. Although I never quite got the right balance between coursework and SICS duties, the educational part of my PhD was the most enjoyable and character building. The skills and patience of the teachers need to be acknowledged by me here. I hope I remembered you all (alphabetically): Daniel Andersson, György Dán, Gunnar Englund, Viktoria Fodor, Anders Forsgren, Mikael Johansson, Ingemar Kaj, Supriya Krishnamurthy, Arne Leijon, Ali Ghodsi, Dan Mattsson, Lars Rasmussen, Mickael Skoglund, Lena Wosinska and Jens Zander.

Funding is critical for the continuity of a PhD, and I have been fortunate to receive financial support from SICS as well as from Vinnova, the EU, Telia AB, Nordunet, SSF and KK-Stiftelsen.

Special thanks are due to Prof. Henning Schulzrinne, the acknowledged expert within IP-based voice communications. It is an honour for me to have Prof. Schulzrinne as an opponent for this work. Also thanks to Doc. Christer Åhlund, Prof. Carsten Griwodz and Dr. Roar Hagen for agreeing to act as grading committee members.

As I have already found my post-doc life in Portugal and I would like to thank the following people for offering me positions, Prof. Manuel Ricardo at INESC Porto, Prof. Rui Aguiar in Aviero, Prof. Luis Correia in Lisbon, Prof. Edmundo Monteiro (plus crazy family of course!), Prof. Fernando Boavida in Coimbra and finally Prof. João Barros in Porto for agreeing to a post-doc position without formally a PhD (here is the dissertation though :-)).

Many people have helped me when things were not the easiest, and I am quite sure I would not be completing the thesis without their professional and unwavering support, in particular Drs. Lars Grahn and Nina Havervall.

To the many friends I met and enjoyed the company of during the years, to name just a few, Iyad, Ehsan, Ali, Jim, György, Ilias, Nacho, John, Evgueni, Henrik, Ibrahim, Petros, Katherine, Berit, Luiza, Kia, Katalin, Adrian, cheeky Ian (another one), Gary and the many others I have surely forgotten to mention.

To my family, especially my Mother, Ray and my fantastic grandmother for all the support and love over the long education. Years ago (I think 1988) I said I wanted to do a PhD and now its nearly done! Last and not least to my devoted and (very) long suffering girlfriend Margarida (alias 'baby Gui'), you deserve the biggest thanks **of all** for accompanying me along the ups and downs of the closing steps of a PhD education. Your crazy cat deserves the final mention in this all too long acknowledgment section for chewing just about every cable I ever owned:-)

I think I'll stop there.

Ian, April 2009.

Contents

Acronyms and terms used in this thesis

Acronyms and terms	Meaning
3GPP	3rd Generation Partnership Project
BGP	Border Gateway Protocol
E-model	ITU objective quality rating
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
EPC	Evolved Packet Core
FEC	Forward Error Correction
GAN	Generic Access Network
GPRS	General Packet Radio System
GSM	Global System
H.323	ITU Internet telephony signalling protocol
ICE	Interactive Connectivity Establishment
IEEE 801.11	Wireless unlicensed Local Area Network standard
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IPTV	Internet Protocol Television
ITU	International Telecommunications Union
LTE	Long Term Evolution
MBONE	Multicast Backbone
MDC	Multiple Description Coding
MOS	Mean Opinion Score
NAT	Network Address Translation
PCM	Pulse Coded Modulation
PESQ	Perceptual Evaluation of Speech Quality
PSTN	Public Switched Telephony Network
QoS	Quality of Service
ROHC	Robust Header Compression
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
SEC	Selective Error Checking
SDP	Session Description Protocol
SIP	Session Intiation Protocol
STUN	Simple Traversal of User Datagram Protocol
TURN	Traversal Using Relay NAT
UMA	Unlicensed Mobile Access
VoIP	Voice over Internet Protocol
WiFi	Commercial synonym for IEEE 802.11 standard networks
WiMAX	Commercial synonym for IEEE 802.16 standard networks

Chapter 1

Introduction

1.1 Internet telephony introduction

Real-time voice communication using IP networks is the subject of this dissertation. The scope of this dissertation is broad and includes several different aspects of real-time voice communication. The effects of the public Internet on telephony sessions have been investigated. Also within our scope is the impact of the access network, and the influence of mobile users. This includes roaming users who can utilise both IEEE 802.11 wireless and cellular networks. The end systems have also been studied and include traditional computers as well as hand-held terminals. Finally, to explicitly include the user expectations in our investigation, we have devised a method to estimate speech quality from real-time network measurements and from off-line processing of sample blocks.

In order to give some background to this dissertation, the upcoming four sections (1.1.1 to 1.1.4) provide a brief description of IP-based voice services. They include four areas in which one encounters the technology - very much from a user perspective. Each section outlines the original impetus for the particular deployment, an introduction to its functionality as well as some possible future directions for each one.

1.1.1 PC-based Internet telephony

From a technological perspective, PC-based telephony came about due to improved CPU performance, permanent and high speed Internet connections, and notably better IP telephony software. Sufficient CPU performance is needed in order to encode the voice for transmission and to decode the received samples. Speech coding is discussed in section 2.4.

Permanent connections are needed to allow incoming calls. Current PC-based telephony software allows calls to be made independently of the local network configuration; this is important as firewalls and routers have caused

setup problems in the past. IP telephony software is now available for essentially all operating systems and hardware combinations including hand-held devices and mobile phones. With this new functionality the personal computer is transitioning from a computing device to a voice enabled communication device. Phone calls are not only limited to computer to computer with PC-based telephony, but using IP to phone gateways, regular phones can also be reached.

PC-based telephony was revolutionised by the popular Skype^{TM} application [30]. It is a cross-platform solution that became successful partly by embracing recent technological developments, and because it provided good, free and easy voice communication. The technological developments it embraced were: Internet-specific speech coding, a firewall bypass solution, a scalable call establishment system, and an intuitive graphical user interface. Skype has continued to add functionality such as inter-operability with the telephony system, a payment scheme, and conferencing capabilities. Recently, the developers have added video and SMS capabilities.

PC to PC communication has become a major success due to Skype and similar applications. The market looks likely to grow by considering the number of Skype online users, see Figure 1.1. As of 2006 VoIP accounted for approximately 20% of the world's telephony traffic of which 4.5% has been attributed to Skype[1]. Therefore, with 80% of the world's telephony traffic still being carried by traditional telephony systems, the migration of voice traffic should further motivate VoIP research.

1.1.2 Broadband Internet telephony

Given the uptake of PC-based telephony, operators realised that similar techniques had a role in cost effective solutions for their voice customers. By leveraging the low cost of high capacity long distance IP links, operators could offer cost effective telephony solutions using the Internet. Different types of operators pursue different strategies: the larger incumbent operators seek to reduce costs, whilst new operators want to enter the voice market with relatively little capital. Both types of operator tend to bundle voice services with Internet access, as the return on providing voice services is falling.

The operator usually provides the customer with a modem into which the customer connects their existing phone and Internet connection. On powering up the modem it establishes the necessary connection, allowing users to make and receive calls using their regular phone. It needs to obtain a local IP address, discover if it is behind a NAT or firewall, and register itself with a server to permit bidirectional media flows. One important phase of this establishment is to locate the correct gateway (see section 2.2).

[1] http://www.telegeography.com/cu/article.php?article_id=15656

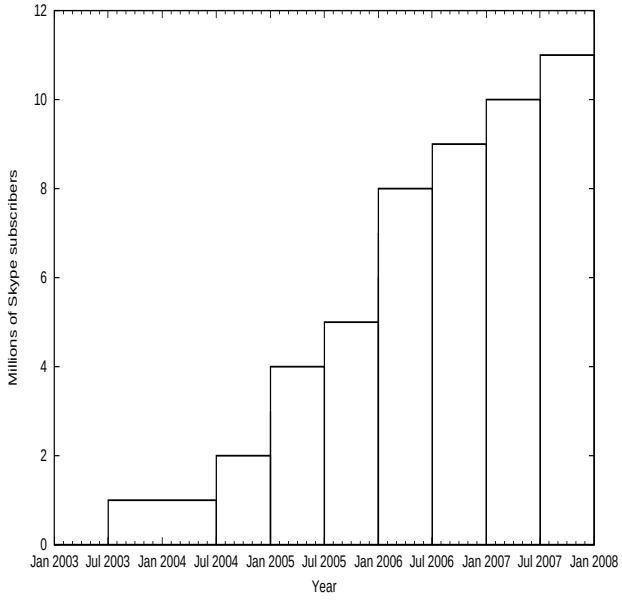

Figure 1.1: Skype usage from August 2003 to February 2008 (source www.wikipedia.org/Skype)

Call records are kept centrally and are used for billing as well as quality monitoring. Subscribers are largely unaware that their voice is partly being transported over the Internet.

Broadband telephony does not require a home computer, making it simpler, more accessible and cheaper than a PC-to-PC solution, and users do not need to be computer literate. Interoperability with the phone system is provided by the operator through a voice gateway. One problem with PC-to-PC solutions discussed in the last section, is that the caller cannot always be identified and located, which is a necessity for emergency calls. Broadband Internet telephony customers on the other hand are registered to an address and thus can make emergency calls.

Broadband telephony is growing, as customers seek to reduce their phone costs, both in terms of lower subscription charges and per minute tariffs. Additional impetus is created by the rising number of homes with broadband Internet subscriptions and (often) bundled voice subscriptions.

1.1.3 IP telephony and the Internet backbone

In the 1990s, research and small-scale tests showed that the Internet was capable of carrying real-time telephony traffic. This was demonstrated with the multicast MBONE transmissions that carried IETF meetings and space shuttle missions. Importantly, the sessions used intercontinental networks, which showed that a business case could be made for wide area real time

voice transport.

A service such as home calling was particularly popular amongst immigrant workers in the United States. Many of the schemes were (and still are) Internet based and prepaid. Traditional phones and local exchanges are used to relay the voice from the regular PSTN network to a gateway, from where the Internet carries the voice over long distance links; the phone network again provides the final leg. Thus the Internet serves as a voice bearer. Some companies saw opportunities in such services, Dialpad and Net2phone were two such examples. Importantly, they both had agreements with the long haul Internet operators. Many thousands of such companies now operate such voice services in most countries of the world.

From the user's perspective, there should be no major quality difference between telephony being carried by the Internet and a regular telephony network. From the operator's perspective on the one hand, it is important that the number of users on the IP network is controlled to avoid overload situations and hence disgruntled customers. On the other hand if a link is being leased for Internet telephony, then it makes financial sense to multiplex as many calls over that link as possible, subject to quality constraints of course. The telephone industry has a highly developed theory (and practice) to allocate calls onto high capacity trunks. This can largely be attributed to one man, A. K. Erlang who produced seminal research contributions from 1909 and onwards. The same theory can be applied to IP networks in order to deduce the allocation of calls per link.

One of the technology remnants from ATM is layer 2 switching: Multi Protocol Label Switching (MPLS) is a carrier technology for IP packets. Basically, MPLS switches labels that are added to IP packets at the ingress of a MPLS network. IP packets that belong to a call are all labelled identically and switched over a dedicated path. Therefore link dimensioning for IP telephony becomes much simpler using MPLS.

The Internet revolution initially bypassed the traditional telecommunications equipment manufacturers and operators. However, the 3rd Generation Partnership Project (3GPP), established in 1998, brought together a number of commercial, organisational and standardisation bodies to work on integrating IP into their solutions for mobile communication. 3GPP has already standardised the use of an IP based core network. Today telecommunication companies are deploying the 3GPP IP Multimedia Subsystem (IMS) to merge Internet technologies with mobile networks. So called 'Release 5' enables operators to upgrade their existing telecommunication equipment and allows a smooth transition to IP technology. IMS is based upon the Session Initialisation Protocol (SIP) which is described in section 2.2.2. The upcoming 3GPP Long Term Evolution (LTE) standard will use IP in both the access and core networks to carry data and voice traffic.

Currently local wireless IP voice services have not reached significant market penetration, as current handsets and infrastructure are dominated

by the telecommunication industry's 2nd and 3rd generation standard solutions. There can be voice quality issues with the current data-centric LAN technologies we have today. The problems are mainly due to coverage and heavy load situations. These are discussed in the next section.

1.1.4 Wireless Internet telephony

Ever more geographically local zones are being established. With the proliferation of dual-mode (local wireless and wide area cellular) telephones, local wireless based Internet telephony could represent an important opportunity for IP-based voice. The France Telecom UNIK service uses dual-mode telephones and a 802.11 gateway, France Telecom quote figures of 25,000 new subscribers per month. In the UK, British Telecom has a similar scheme and T-mobile will launch their own service in the US during 2009. The IEEE 802.11 standards are the current technology preference for local wireless access.

As far as voice traffic is concerned, there are two broad usage scenarios within local wireless networks. One is to only use the local wireless technology; voice calls are not continued should the user move from the coverage area. Therefore movement is restricted to within the coverage area. Note however that the coverage area may comprise several access points allowing some geographic area to be covered within one administrative domain. Further deployment and new technologies will allow for greater coverage in the future. Collectives are being formed based upon coverage and financial incentives to set up and share wireless networks, e.g. the Fon and Skype Zone initiatives.

Voice quality can suffer if there are radio coverage problems, interference from external sources, and excessive network load. The range for good quality varies from a few metres to a hundred meters depending on the equipment in use, obstacles, interference sources, and so on. Therefore the second scenario is to switch calls between the local wireless and cellular infrastructures in order to provide call continuity outside the coverage area of the wireless LAN. As mentioned, mobile phones and PDA's are now available with both cellular and 802.11 interfaces. This provides an option for switching to the cellular network when needed. Alternatively, if local wireless coverage is detected during a cellular call, a switch to the local network is possible, thus freeing cellular resources and potentially avoiding the cellular operator's tariffs. Entering a home or office area are typical scenarios in which a cellular call could be transferred to the local 802.11 network. The procedure of switching an ongoing call from one technology to another is known as a handover or handoff. Ideally the user should be unaware of the change, if this is the case it is known as a *seamless* handover. The current technological barriers for seamless handovers are the configuration and connection establishment mechanisms rather than the switching of the voice

stream. Switching a voice stream means receiving two parallel streams to the same terminal over different networks. Once running in parallel to the terminal, the initial stream can be stopped and the new voice stream played to the caller instead.

As we are interested in maintaining call quality, the timing of handovers from the WLAN to the cellular network is important. In the case of radio problems there might be insufficient time to initiate and start a call to the cellular network. In the case of handover due to the onset of congestion, the handover success depends on the rates of the other flows. This is due to the time needed to estimate the call quality and if need be, to initiate a cellular-based call. In the other case where a user would move out of the coverage area, there should be time to schedule the handover. The speed and path of the user movement can be tracked to estimate whether the user is moving out of coverage. In this case there is a design tradeoff: To maintain connectivity in the coverage area as long as possible to minimise the frequency of handovers on the one hand, or to reduce the probability of poor quality and switch early on the other. Therefore more conservative or aggressive switching algorithms can be envisaged.

Generic Access Network (GAN), formerly known as UMA (Unlicensed Mobile Access), is one possibility to provide seamless roaming between local and wide area networks [31]. GAN allows voice, data, and IMS/SIP applications to be accessed from a mobile phone. The operation of GAN is as follows: Once a local wireless network is detected (e.g. Bluetooth or 802.11) the handset initiates a secure IP connection through the local network to a gateway in the operator's network. A GAN server makes the handset appear as if it were connected to a new base station. Thus, when the handset moves from a cellular to a 802.11 network, it appears to the core network as if the handset is simply associated with a different base station. There is GAN support for 2nd and 3rd generation cellular technologies.

1.1.5 Summary of the introductory sections

Apart from the obvious human need to support real-time person-to-person communication over geographic distances, it is hopefully clear from the last four sections that Internet telephony has a permanent position in modern communication networks. As voice is a real-time conversational service, there are strict requirements on the end-to-end quality characteristics that the telephony operator must provide in order deliver a successful and robust service. We will now look at the task of fulfilling these requirements as research topics within this dissertation.

1.2 Dissertation outline

This sections gives the motivation, problem statement, methods, conclusions and potential topics for future research. There are additionally short descriptions of the research contributions of each publication and the individual contributions of this dissertation's author.

The second chapter presents the major IP telephony building blocks. A short description of the path voice samples take from speaker to listener is given. This is to illustrate the typical processing that voice samples undergo. Subsequently, sections on signalling, speech coding, firewall traversal, voice Internetworking and human tolerances to digitised speech are elaborated upon further.

The third chapter of the dissertation concerns Internet telephony from a quality perspective. We go through some of the mechanisms used to assess speech quality, including measuring and estimating quality, plus an overview of two ITU proposals for objective speech quality assessment.

The fourth chapter is a research literature review from a historical perspective. It is divided chronologically, into episodes of the development of Internet telephony from early packet switched experiments to world-wide deployment.

The appendix of the dissertation is composed of the nine published papers. The structure of this dissertation is shown as an illustration in figure 1.2.

1.3 Dissertation motivation

In the previous sections we have seen various settings for IP-based voice in telecommunications systems. Although each has its own particular challenges when it comes to providing acceptable quality for its users, we can formulate a unifying motivational statement for this work: To carry real-time voice from speaker to listener with acceptable quality under a range of operating conditions.

This statement can be further subdivided into seven motivating reasons for this research.

Current relevance: Real-time voice is still the most efficient media to carry information quickly and unambiguously from person to person. Although email, instant messaging and SMS have become popular recently, the unequivocal importance of real-time voice communication remains.

Network challenges: Using IP networks to transport real-time voice can be challenging. The complex nature of bulk IP traffic makes a complete understanding of the aggregate behaviour difficult, especially when viewed

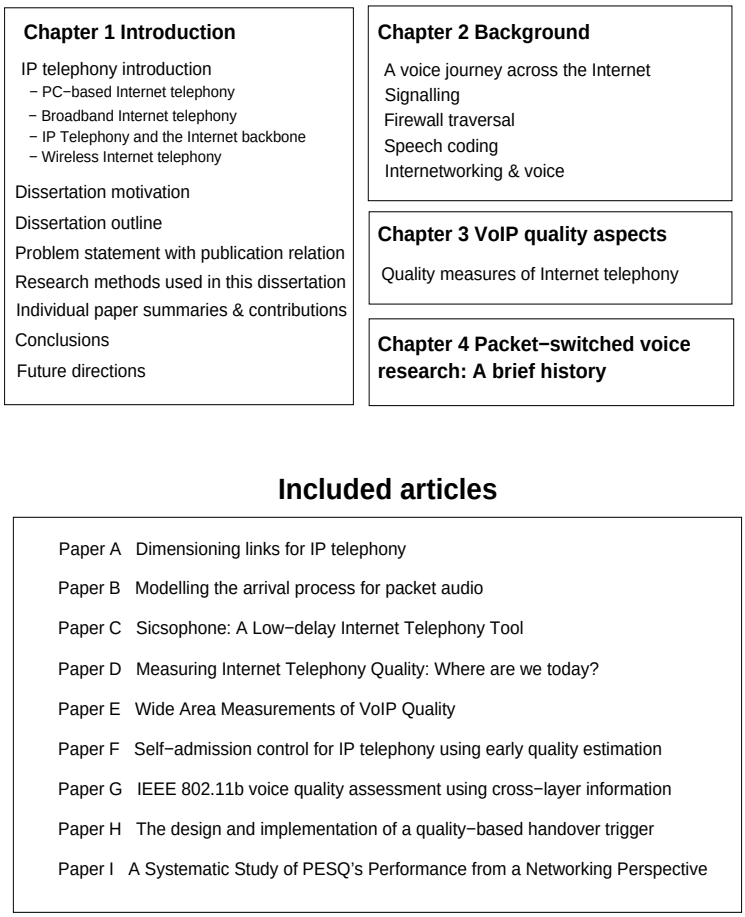

Figure 1.2: Dissertation structure

from different time scales. Where voice data is multiplexed with many data flows, the received speech sequence usually does not resemble the transmitted sequence. Traffic demands vary on the Internet to some degree according to popular applications and services, therefore there is no fixed target to design for. In addition to the traffic, there are differences in the operating environments, such as fixed and wireless access networks or transit and backbone networks. Despite the known user requirements for voice, the conditions under which it is delivered leads to a complex problem.

Implementation feasibility: New solutions can be introduced into IP networks. The relatively simple IP programming interface facilitates novel and innovative solutions. Whole or partial solutions are implementable using approximately 20 library functions. This is in stark contrast to the telephony system which requires detailed specialist knowledge for application development.

Subjective assessment: It is possible to assess perceptually the success or failure of IP-based voice research. In subjective assessments real people listen and indicate scores according to the quality of the speech. There are two forms of subjective tests, comparative or absolute. Comparative tests indicate the perceptual gain with and without improvement. Absolute tests simply ask whether the quality delivered is acceptable without a comparative signal. The major disadvantage with subjective tests is that real subjects are required, the trials should be conducted according to expensive standard procedures and eventually test subjects become tired. There are alternative objective measures, which we have used in our research, but are clearly less accurate.

Understanding broader traffic issues: There are two aspects to be considered in a broader sense. First, since the voice data is generated as a (nearly) periodic stream, it acts effectively as an active probe along the network path. The stream can reveal useful information of the path conditions by reporting properties such as the loss and delay distributions. Second, investigating the effect of large data volumes on "thin" voice streams may indicate what measures need to be taken to implement protection for delay sensitive traffic. In some respects understanding the behaviour of this mixed traffic is the key to better network planning. If network mechanisms are to be introduced to maintain balance, predictability and quality of service for voice and other time sensitive media, then the interplay of mixed traffic types should be investigated.

Terminal heterogeneity: Ultimately the voice must be replayed for a listener. Minimally, the timing information must be restored to produce the original speech pattern and (optionally) lost frames masked. The functionality of the receiver depends very much on the type of hardware, operating system, computational power, battery capacity, and the network to which the terminal is connected. The motivation of this work therefore, with respect to terminal heterogeneity, is that each solution needs careful tailoring for a particular hardware/software combination.

1.4 The problem statement and its relation to the publications

Let us begin with a non-problem. In *principle*, capturing, processing, transmitting and receiving real-time voice samples that use an IP infrastructure is non-problematic. Voice samples are captured, coded and sent at constant intervals. Samples are batched together as packets, addressed and sent across shared access, transit and backbone networks. The packets are received and are buffered in order to provide a continuous stream of samples

Paper	Title
A	Dimensioning links for IP telephony
B	Modelling the arrival process for packet audio
C	Sicsophone: A low-delay Internet telephony tool
D	Measuring Internet telephony quality: Where are we today?
E	Wide area measurements of VoIP quality
F	Self-admission control for IP telephony using early quality estimation
G	IEEE 802.11b voice quality assessment using cross-layer information
H	The design and implementation of a quality-based handover trigger
I	A systematic study of PESQ's performance from a networking perspective

Table 1.1: List of papers in the dissertation

for an application. The samples are removed from the packets, the timing restored and passed to the operating system for playout. The purpose of this brief explanation is to illustrate that no extraordinary processing needs to be performed in the absence of network, or end system abnormalities. In other terms, well dimensioned networks and capable end systems should be sufficient for ample quality voice communication.

The problem statement therefore is as follows: *Delivering a real-time good quality voice service over multiservice, multiplexed IP communication paths supporting stationary and mobile users using heterogeneous terminals.* Using the publications included in this dissertation (see Table 1.1), we will now discuss the problem statement and their relation.

In paper **A** we look at how to allocate resources for a single service voice network. The problem to solve is how to regulate the number of calls entering a system so that acceptable user quality can be delivered. The paper considers an IP network in which only voice is carried, somewhat similar to a telephone network. In relation to the problem statement we are looking at the *multiplexing effects* of IP-based voice streams.

The above scenario may be thought of as somewhat naïve in the IP case. In practice the networking (and computing) resources are often shared, thus disruptions in the voice stream are possible. Therefore paper **B** addresses the issue of modelling packet disturbances in order to reconstruct the *variance distribution* as observed by the receiver. Having a model of the variance helps the receiver in making more informed decisions on what actions to take as packets arrive. Modelling the variance distribution is complicated by the fact that packets can be lost and that silence periods are introduced into the stream when the speaker is quiet. For the model, it is assumed that the network delay distribution is estimated, measured, or indeed known.

Replaying voice streams on real end systems is the topic of paper **C**. This means buffering the arriving packets at the end system and reconstructing the original timing from the RTP packet header information. Not only should the process be accurate, but with the lowest possible delay (and loss). In this work, we provide a method that utilises the operating system and hardware efficiently. We have implemented, tested, and measured a new

approach to end system design for voice streams. In relation to the problem statement, we are addressing the problem of good quality communication.

To gain insight into the real-world aspects of Internet telephony we have undertaken two large wide-area measurement experiments. By large we mean using hundreds of generated calls in the first experiment and thousands in the second. Analysis of these experiments are described in papers **D** and **E**. The problem is to obtain representative measurements from the end systems we had access to. One issue with measurement tasks (generally) is to *completely* anticipate the needs before the upcoming analysis. As well as our own measurement experiments, we were aware the data would be used in related investigations, both by us (papers **B** and **F**) and by other researchers. Therefore acquiring all the necessary information for related studies requires a fair amount of foresight. As a simple example executing `traceroute` before and after each session might help backtrace why a session exhibited abnormal behaviour. As we have taken two partially intersecting sets of measurements taken four years apart, we would like to compare the results for any trends. In relation to the problem statement, we are studying the multi-service nature of Internet traffic.

Paper **F** explores the idea of terminating sessions early when poor quality can be predicted. This can be seen as a problem of self-admission, implying that a call should not continue if an estimate of the call quality is below a quality threshold. Using data from paper **E**, the problem becomes how to determine this threshold, as well as the time needed to reach a decision. In relation to the problem statement above, this paper addresses actions to be taken when conditions deviate from an acceptable operating range.

Wireless and mobile IP systems have their own set of associated challenges which can impact on the voice quality. In wireless systems, stochastic link conditions is one inherent factor. In addition, the radio frequency bands used by 802.11 interfaces are not licensed, and hence not regulated, so interference can occur from other devices. We have investigated VoIP quality over 802.11 networks using cross layer information in paper **G**. In relation to the problem statement we are considering the mobile, and hence wireless, users.

One solution is to use the 802.11 network where possible, but to handover a call to the cellular network when the link conditions are insufficient to support good quality as stipulated in the problem statement. How to schedule this handover has been addressed in paper **H**. Real-world voice handovers typically need time to initialise a parallel technology to switch to. As calls to the public phone network take in the order of five seconds to setup, estimation of deteriorating quality conditions in the 802.11 network must anticipate (at least) this interval ahead of the handover. The relation of this work to the problem statement is in the *heterogeneity of the systems* and providing *good speech quality* to the users.

Ultimately users must be satisfied with the quality of the voice recre-

Technique	Paper
Mathematical modelling	A, B
Discrete event simulation	A
Implementing proof-of-concept applications	A, C, H
Active measurements	E, D, G
Statistical analysis	B, F, I
Subjective user tests	I

Table 1.2: Summary of research methods used within this dissertation

ated from the incoming data stream. Missing parts of a sentence or lost keywords can easily lead to unintelligible phrases. The challenge of paper **I** is to understand how packet losses effect speech intelligibility. Our goal was to produce an estimator that can monitor packet losses and output a simple indicator of the speech quality. To be of any real practical use, our evaluation should correlate with that given by a person who listens to the same sequence. The advantage of having an objective measure is that the system can react to what it thinks is poor quality speech being delivered to the user (or ideally before). The relation of this work to the problem statement is *good quality* and *mobile users*.

1.5 Research methods used in this dissertation

We have used a number of different techniques to solve the problems discussed in the last section. The research in this dissertation focuses on real-world problems concerning quality aspects of real-time packetised voice. The techniques used and the paper letters are shown in Table 1.2. The upcoming paragraphs step through these methods one by one and state in which work, and to what degree, the methods were used.

Mathematical modelling: Within this dissertation, we model the statistical multiplexing of telephony calls in paper **A**. By modelling the multiplexing we can produce a tractable approximation of a telephony system consisting of packet streams from multiple callers arriving at a single queue. In this model the number of calls is governed by a Markov process and each packet stream as a Poisson process. The resulting flows at a multiplexer constitute a Markov Modulated Poisson Process (MMPP). The role of the model is to form a tractable approximation of the number of flows that can be allocated to a given link capacity, and the size of the buffer at the multiplexing point.

In paper **B** we model the arrival process of a single IP telephony stream at a receiver. We consider two types of delays for a given packet: the delay caused by waiting behind previous telephony packets and the delay

introduced by cross traffic along the same path. The arrival process is modelled as a discrete time Markov chain. The function of the model is to reveal the delay distribution of the packets at the receiver.

Discrete event simulation: Discrete event simulation is used to model the propagation delay of the individual packets from multiplexed voice sources in paper **A**. Each packet is traced from source to destination. The simulator counts packet loss at the multiplexer. `ns-2` was used as the simulation framework and the goal of the simulation was to confirm or deny the accuracy of the MMPP model described above and an implementation described below.

Implementing proof-of-concepts: As well as the obvious working software, we have used a proof-of-concept in paper **A** to verify the accuracy of the model and simulation. The working implementation shows whether the theory and practice match, and whether the solution can be deployed into an operational network with some confidence. Proof-of-concept implementations also show which parts of the model are missing, either by design due to abstraction, or simply not accounted for in the problem formulation.

In paper **C** we have implemented a voice playout strategy to reduce the delay incurred by a VoIP receiver. The solution was implemented on a standard PC running different versions of the Windows operating system. The basic idea is to avoid copying the data from the operating system, to the application, then back to the operating system for playout. DirectX now provides similar functions to perform copying in this manner. The role of the implementation is clear, to test and measure the improved playout mechanisms.

In paper **H** we implemented an automated handover mechanism on a PDA running Windows CE. We estimate the call quality in the terminal based on network measurements and signal a third party application that the current call should be transferred from the 802.11 network to the cellular network. The handover was triggered when the quality fell below a quality threshold. Our implementation allowed automatic roaming from 802.11 to GSM networks. The goal of the implementation was to show proof of concept, as well as to judge differences in the speech quality at the time of handover.

Active measurements: Active in-band measurements have been used to sample the path properties during our standard call. The main goal of the measurement work was to report on the suitability of diverse paths with respect to real-time voice. Although limited to academic sites, we chose a wide range of path diversities in order to generalise the results as best we could. One additional reason for conducting the measurements was at

that time (1998 and 2002), no extensive public measurement data was freely available. The measurement work forms the core part of papers **D** and **E**. Some comparison between the two data sets was done to determine whether the quality improved or deteriorated between the measurement periods. We used a modified version of the tool described in paper **C** for the measurement work.

We made a comprehensive evaluation of 802.11 networks using active measurement techniques reported on in paper **G**. Since we had control over the network we were able to perform systematic tests starting from simple (line-of-sight ad-hoc) to complex (infrastructure with competing traffic) experimental setups. The main objective of the active measurements in this case was to capture and quantify the stochastic behaviour of the 802.11 network with respect to voice traffic. A secondary objective was to utilise cross-layer methods that are well suited to voice over wireless applications as demonstrated by the cellular solutions.

Off-line analysis: The active measurements have been used in our off-line analyses. Paper **B** modelled the arrival process of individual voice streams, where measurements from paper **E** were used to validate the inter-packet predictions of the model. Paper **F** used the measurement data from paper **E** in an attempt to estimate which calls would yield poor-quality conversations from the initial seconds of a call. The information from the rest of the call showed whether the decision was indeed correct or not. In paper **I** we used a tool standardised by the ITU (PESQ) to estimate the subjective effect of packet loss on standard eight second voice samples. Our results were used to map network losses to an approximation of the subjective quality. Due to the complexity of the PESQ algorithm in terms of the signal processing, such tests have to be done off-line.

Subjective user tests: In paper **I** we used test subjects to indicate a quality rating for pre-recorded speech samples. The subjects listened to several eight second degraded samples and rated their opinions on a nine point scale. We used 11 test subjects and set up the tests according to the P.862 ITU recommendation [135]. The objective of this recommendation is to ensure that tests are conducted systematically, with an appropriate test duration, warm up tests, deafness tests and so on. The goal of this work is to compare the subjective results with those given by PESQ. The role of such experiments within networking research is often underplayed where the results can be judged by real users.

We also used subjective user tests in paper **H**, where the quality of voice was rated before a handover from the 802.11 to the cellular network. Where the quality started good and ended up poor and a handover was suggested, we recorded this event as a positive result. Where the quality

started good and remained good, and a handover was not suggested we also considered as a positive result. In the two other situations the handover estimation was deemed a negative result. The total number of positive results, in comparison with the sum of positive and negative results gave the performance of our handover algorithm.

1.6 Paper summaries and contributions

Paper A

Bengt Ahlgren, Anders Gunnar (née Andersson), Olof Hagsand, and Ian Marsh. Dimensioning links for IP telephony. In *Proceedings of the 2nd IP-Telephony Workshop*, pages 14-24, New York, USA, April 2001.

Summary: The number of IP telephony calls that can be admitted to access networks is addressed in this paper. Link dimensioning based on packet loss is one method for dimensioning links for high utilisation of networking resources whilst providing acceptable user quality. Using this approach we also show how to select router buffer sizes. We validate and compare our approaches using a mathematical model, a discrete event simulation, and a laboratory-based implementation.

Contribution of this work: The contribution of this work is a planning tool for use in dimensioning networks for voice traffic. We have established a relationship between the important parameters of a packet voice network: namely the speech coding, the link capacities, the number of users, the buffer sizes, and the acceptable loss rates.

My contribution: The original idea to perform such a study was mine. I implemented most of the testbed environment and the traffic generator. Within the project I supervised a masters student, Anders Gunnar (née Andersson), who implemented the MMPP model in Matlab and corresponding simulation scripts in `ns-2` [100]. Anders was co-supervised by Professor Ingemar Kaj at Uppsala university. We were assisted by Henrik Abrahamsson, Bengt Ahlgren, Olof Hagsand and Thiemo Voigt. I co-wrote the paper with Anders and presented it.

Paper B

Ingemar Kaj and Ian Marsh. Modelling the Arrival Process for Packet Audio. In *Quality of Service in Multiservice IP Networks*, pages 35-49, Milan, Italy, February 2003.

Summary: In this work, we model the arrival process of voice packets at a receiver. The assumption is that the original packet spacing has been disturbed by bulk data transfers and queuing behind packets of the same stream. The solution, based on a Markov model, models the delay variation of the speech packets. The packets are assumed to be subjected to network delays when travelling from source to destination. The waiting time in intermediary buffers is assumed to be exponentially distributed. The use of such a model allows silence suppression and packet losses to be incorporated; as they are independent of the network induced delay variation.

Contribution of this work: The contribution of this work is a model for the packet audio arrival process. A simple method to estimate packet loss based on observed interarrival times is also given, independent of whether silence suppression is used or not. The model was verified by measurement data.

My contribution: The idea was jointly conceived. My contribution was the measurement data and validation of the model data. I also wrote several tools to process the data. I co-wrote and presented the paper.

Paper C

Olof Hagsand, Ian Marsh, and Kjell Hanson. Sicsophone: A Low-delay Internet Telephony Tool. *IEEE 29th Euromicro Conference*, Belek, Turkey, September 2003.

Summary: All VoIP systems terminate with a receiver. It can be a PC, hand-held terminal, or phone. The terminal has an important role in the overall system performance. For the PC case, we look at how to reduce delay through a novel receiver buffering scheme. The solution uses the low-level features of audio hardware and a specialised jitter buffer playout algorithm. Using the sound card memory directly eliminates intermediate buffering. A statistical-based approach for inserting packets into the audio buffers is used in conjunction with a scheme for inhibiting unnecessary fluctuations in the system. For comparison we present the performance of the playout algorithm against idealised playout conditions. To obtain an idea of the system performance we give some mouth to ear delay measurements for selected VoIP applications. The proposed mechanism is shown to save 100's of milliseconds on the end to end path.

Contribution of this work: The contribution of this work is a sizable reduction in the delay incurred by the VoIP end system. Although many researchers have looked at optimising and reducing jitter buffer sizes, many

do not implement their ideas in a real system. An important byproduct of this work is Sicsophone, a fully functional VoIP application.

My contribution: I wrote the RTCP part of Sicsophone. I performed comparisons between the playout delay of Sicsophone and the optimal playout delay. I co-wrote and presented the paper.

Paper D

Olof Hagsand, Kjell Hanson, and Ian Marsh. Measuring Internet Telephony Quality: Where are we today? In *Proceedings of IEEE Globecom: Global Internet*, pages 1838-1842, Rio De Janeiro, Brazil, December 1999.

Summary: Users of Internet telephony applications demand good quality audio playback. This quality depends on the instantaneous network conditions and the time of day. In this paper, we describe a scheme for measuring network quality and motivate the development of a new metric for VoIP, *asymmetry*, to include into quality reports.

Contribution of this work: In 1999 we reported on the findings of our first VoIP measurement study. As far as we are aware of, the jitter and asymmetry results were new within the VoIP community. The number of downloads of the data from a COST Action web site exceeded 100.

My contribution: The idea, measurements, and paper were done by me. I wrote and presented the paper. The Sicsophone tool used to conduct the measurements was originally written by Olof Hagsand and Kjell Hanson with some modifications by me for the measurement work.

Paper E

Ian Marsh and Fengyi Li. Wide Area Measurements of VoIP Quality. *Quality of Future Internet Services*, October, 2003, Stockholm, Sweden.

Summary: We have investigated the network characteristics of loss, delay and jitter for VoIP streams that are transmitted over diverse Internet paths. Based on over 24,000 sessions, taken from nine sites connected in a full-mesh configuration, we reported on the average quality that can be expected by a user. The VoIP quality was acceptable for all but one of the nine sites we investigated. We also concluded that VoIP quality had improved marginally since the previous study in 1999 (paper **D**).

Contribution of this work: The contribution of this work is a comprehensive report on the quality of Voice over IP in 2002. We defined the quality in terms of the one-way delay, loss, and jitter. For three of the sites, we have been able to compare the quality from 1999 to find some trends in VoIP quality. More than 500 downloads of the data have taken place since they were made available. The data has been used papers **B** and **F** within this dissertation.

My contribution: The idea to improve on the measurements from 1999 (Paper **D**) was mine. I advised a masters student, Fengyi Li, to perform the measurement tasks. Further modifications of Sicsophone were done by me. I wrote a tool to process the measurement data. We jointly wrote the paper based on Fengyi Li's master thesis [87], I presented the paper.

Paper F

Olof Hagsand, Ignacio Más, Ian Marsh and Gunnar Karlsson. Self-admission control for IP telephony using early quality estimation. In *4th IFIP-TC6 Networking*, Athens, Greece, May 2004.

Summary: The idea is to use packet loss statistics from paper **E** to potentially identify poor quality calls given only the initial seconds of a call. The application is a self-admission control scheme, which will continue or terminate a call depending on a quality threshold. The threshold is determined by the acceptable loss rates of the speech coding used. If sessions themselves can determine whether entry into a system is worthwhile, given the early loss rates, then system resources and user frustration can be avoided.

Contribution of this work: The contribution of this work is a self admission control for IP telephony. The scheme does not require any network support or external monitoring schemes.

My contribution: My role in this work was in the initial discussions and providing the measurement data. Some filtering of the data was needed to begin the work, hence I wrote the initial version of the data parsing tool. We jointly authored the paper.

Paper G

Ian Marsh, Juan Carlos Martín Severiano, Victor Yuri Diogo Nunes, and Gerald Q. Maguire Jr. IEEE 802.11b voice quality assessment using cross-layer information. In *1st Workshop on Multimedia over Wireless*, Athens, Greece, April 2006.

Summary: The conditions that VoIP users can encounter in 801.11 networks is covered in this paper. It is measurement based and takes a methodological approach to understanding quality variations in 802.11b networks. We started with simple point-to-point VoIP experiments to determine the delays associated with the terminals and operating systems.

We progressed onto 802.11 infrastructure mode using line of sight and indoor measurements. Next non line of sight experiments were conducted and again re-conducted in the presence of competing TCP traffic. Some simple, but effective, mechanisms were proposed to maintain acceptable VoIP quality using 802.11 networks. We used the Sicsophone tool amended with modules for obtaining the MAC layer retransmissions and data rates.

Contribution of this work: The contribution of this work is a comprehensive study of 802.11b networks as far as voice is concerned. This includes the methodology we employed plus utilising cross layer techniques to obtain our desired results. Many of the lessons we learned were put to use in paper **H**.

My contribution: The ideas for the project were mine. Most of the work was carried out by two masters students, Severiano and Nunes, working on the MAC/IP layer interactions and on the IP/application layer interactions respectively. Gerald Q. Maguire Jr. co-supervised the students. We all authored the paper.

Paper H

Ian Marsh, Björn Grönvall and Florian Hammer. The design and implementation of a quality-based handover trigger. In *5th IFIP-TC6 Networking 2006*, Coimbra, Portugal, May 2006.

Summary: In this work we looked at the conditions under which an ongoing call could be migrated from a 802.11 to a cellular network without perceivable loss in quality. We performed measurements on the 802.11 network in order to make workable predictions of the call quality. We implemented our solution on a hand-held terminal and performed 100 handover test trials of our handover mechanism.

Contribution of this work: The contribution of this work is one part of a fully working system that allows calls to be migrated from a 802.11 to a GSM network automatically.

My contribution: Björn Grönvall and I jointly conceived the initial idea and jointly performed the base experiments on which the automatic trigger

was designed. We co-implemented the solution. We also integrated our solution into software developed by Optimobile AB. Florian Hammer helped in the PESQ assessment of packet loss. Björn Grönvall and I wrote the paper and I presented it.

Paper I

Martín Varela, Ian Marsh, and Björn Grönvall. A Systematic Study of PESQ's Performance from a Networking Perspective. *Proceedings of Measurement of Speech and Audio Quality in Networks*, Prague, Czech Republic, May 2006.

Summary: The basic idea is to have a general function which maps losses into estimations of the quality due to packet loss. Using standardised samples distorted by network losses, we could utilise PESQ processing off-line to map packet losses into quality ratings over a range of operating conditions. We verified our results with real test subjects. We also compared the single sided measure (ITU P.563 [67]) to our own findings.

Contribution of this work: The contribution of this work is a real-time single-sided metric for estimating speech quality. A systematic study of the behavior of PESQ as a function of losses has also been performed. Also the variability of PESQ ratings under several different test conditions has been conducted. The PESQ ratings were compared to subjective scores for a range of bursty losses.

My contribution: I worked jointly with Martín Varela and Björn Grönvall on the idea. We conceived the idea together. Martín was responsible for most of the scripts, whilst we both analysed the data. The paper was jointly authored.

1.7 Conclusions

This dissertation addresses selected topics within real-time voice communication. Our focus is on the *quality aspects* of voice communication, since poor quality often leads to user dissatisfaction. The techniques presented in this dissertation attempt to solve the research problems independent of network QoS efforts.

Each of the publications draws similar conclusions, that is, reasonable quality Internet telephony can be offered, provided that the whole system is carefully engineered. This implies the introduction of mechanisms to preserve the subjective quality when impediments are, or are about to, occur. Some of the conclusions from our research are as follows: The network load

should be controlled for links that carry real-time voice. This means providing and dimensioning links with sufficient capacity, or alternatively, restricting the admission of voice calls to heavily loaded links. The monitoring of network conditions, in particular loss, should be used to signal potential quality problems on particular paths. We have presented a solution where the end system can do the monitoring where such network functionality is absent. Should we require earlier indications of impending problems, tracking the network delay or jitter at the end system can be investigated. This technique has been used in our handover studies, where several network parameters have been combined in order to schedule a handover. Continuing in the wireless case, we have proposed mechanisms for maintaining quality by switching to lower data rates, or even switching to an alternative technology where available.

Since the scope of this work is broad, we have taken different cuts through IP telephony research by looking at the access and backbone networks, using modelling, simulation and experimental techniques; we have considered both fixed and wireless networks using subjective and objective quality tests to obtain the most appropriate solution for a particular problem. We have also looked at systems with and without background traffic, used real-time and off-line techniques, and finally applied cross layer approaches that combine normally separated layers of the protocol stack.

The main contribution of this work is a near-complete system study concerning quality aspects of an Internet telephony system. We have looked at a number of different methods to enable adaptable solutions for maintaining acceptable quality. We have often found that relatively simple changes can lead to substantial user quality gains.

The tangible outcome of our research has been a number of software tools. These include an IP based voice measurement package, a handover algorithm for wireless terminals, a VoIP traffic generator and a PESQ processing package.

1.8 Future directions

Plenty of challenges remain within the area of IP-based voice quality. We will consider each one in the context of the research done within this dissertation, and later on discuss broader topics outside the scope of this work.

In-dissertation issues: In the area of network provisioning, a macro level investigation needs to be conducted on the suitability of the MMPP model for dimensioning tasks on an Internet scale. Our investigations were done and verified for links only up to 1.5Mbits/s. Therefore, one (ambitious) theoretical study could be to investigate migration of the world's telephony traffic onto the Internet. This would partly include capacity studies of the

existing voice traffic, separating voice traffic from TCP flows, and estimating the future demands of voice on the Internet, thus scaling up the dimensioning work to much larger network capacities.

On the individual flow level, research should be done on understanding the network delays for voice packets over different operating conditions and network types. Due to the increase of bandwidth-heavy applications such as P2P traffic and video streaming, the conditions for voice traffic needs to be reinvestigated. As far as the network is concerned, the arrival process for VoIP packets over wireless links should be reexamined. One reason is access to the medium is distributed, allowing multiple flows to become multiplexed at the first hop. Finally the concept of backoff timers in CSMA protocols has not been included in our model.

More work can be done on hand-held terminals to support voice applications. This is because smartphone type terminals currently offer insufficient voice quality on 802.11 networks. Essentially this is because terminals are computers and the 802.11 protocols have been designed for data transmissions. Furthermore, the networking interfaces are commodity items and do not provide sufficient handles for voice application designers. We have said earlier, voice applications on IP networks need careful engineering. The telephony side of some smartphones is separate and has a highly integrated system using techniques such as joint source and channel coding. Voice application writers do have the access to such technologies, they simply have a strict layered protocol stack to interface to. In the specific case of 802.11 networks, application writers would benefit (at least) from access to the MAC retransmission counters, precision RSSI signals, data rates and near instantaneous bit error rates at the link layer level.

As far as active measurements are concerned, additional investigations should target home users to include their usage patterns. This includes 802.11 based networks and telephony. Coordination and collaboration with ISPs would be beneficial in order to obtain a broader sample set of users, as well as important data on the network operational status. Some cities operate open 802.11 networks which could be instrumented to obtain better operational status. As the 4th generation networking technologies are almost upon us, investigations of voice over the newer radio access technologies (e.g. OFDMA) and the Evolved Packet Core (EPC) would surely be desirable for a new look at capacity planning on telecommunication networks.

We believe there is still much research to be done in the voice handover area, including monitoring the network conditions at the handset. As we have eluded to earlier, tight integration achieves the best results and in the case of dual-radio phones, prediction of impending problems is the key criterion. Not included in this research is the possibility to make use of tracking i.e. estimating the position or path of the user. This would greatly influence the decision of whether to switch a call to an alternate technology.

Further work needs to be done on objective quality assessment tools.

While PESQ and the single-sided measure methods exist, improvements can still be made. From our experience the performance of these methods deviates as the loss process becomes more correlated. Naturally, it is difficult to adjudge a series of samples with missing segments, with or without the reference signal, nevertheless, such loss processes are reality on many wireless networks today. Also one would like to include delay into the assessment, as current methods are loss-based only.

People adapt to delays, by less frequent interruptions in the conversation for example. Conversational quality models have been proposed, however their accuracy is still not clear.

Broader issues: Moving onto just one topic outside of this dissertation, we believe that higher fidelity telephony should be available in the near future. Although the technology for transporting bits has improved, the media stream itself has not changed since the introduction of 64 kb/s voice many decades ago. From the user's perspective the voice quality of a 13 kb/s stream is actually worse than that of traditional telephony. However, we are prepared to pay this cost in order to have mobile telephony, and, of course, the operator can squeeze more calls out of the system without substantial investment.

The drive to reduce bitrates for calls has been to multiplex more calls onto capacity constrained links. However, as ever more capacity is becoming available both on the cellular and Internet technologies, the time is right for a new type of voice experience. Therefore, one example would be to use *higher* fidelity than we are currently used to. This may be stereo voice, and would require headsets, but many mobile users already use such devices to listen to music.

Going one step further is 3D telephony. This will enhance the experience at the listener through capturing binaural signals at the speaker, optionally rendering them in 3D space, and replaying the enhanced signal at the listener. Capturing the signals at the speaker can be done by placing small microphones on the outside of the headsets, somewhat similar to what noise cancelling headsets do today.

Steps such as these would represent a new domain for telephony that has been thus far the preserve of specific environments such as audio conferencing. 3D telephony is very much under investigation, however significant challenges remain, particularly in the domain of noise cancellation, either at the sender or receiver, or both.

Chapter 2

Background

This chapter consists of two parts. The first is a short description of the path that voice samples take from a sender to a receiver as part of a VoIP system. The second part contains sections on some of the important building blocks of IP telephony: signalling, firewall traversal, speech coding and IP networking.

2.1 A voice journey across the Internet

Figure 2.1 shows the processing components (as blocks) typical for a stream of voice IP packets. The voice is captured by a microphone, sampled, digitised, and encoded into a format chosen by the application. Typically a voice frame is of 20 ms duration and contains 160 voice samples, where each sample is 8 bits of information sampled at 8000 Hz.

FEC/MDC (Forward Error Correction/Multiple Description Coding) can create redundant samples from the existing samples. The redundant samples are transmitted with a time shift from the original samples to reduce the probability for losing both the original and redundant data. The encoded voice is then packetised which means gathering the samples into one transmission block. Addressing information is pre-pended to the block which includes RTP, UDP, and IP headers. The packet is sent onto the local network via a network interface. A link local frame header is appended for each link traversed on the path.

The packet traverses one or more networks where multiplexing occurs. Once the packet reaches the receiver, the headers are removed and any FEC or MDC that was applied can be used to recreate lost packets. The packets need to available for decoding in continuous blocks, therefore they are buffered and timing information in the RTP information used to generate the sequence. The application can also take action if the packet loss protection was not sufficient, voice frames can be created using a technique called packet loss concealment (PLC) where lost samples are masked by creating

approximations of the lost samples from those received. Finally the voice samples are transferred to the end terminal's audio output device (shown as speaker). In some cases the speech decoding and loss concealment may be combined in one algorithm.

Internet telephony can be broken into two distinct phases: signalling and voice transfer. The signalling phase is responsible for the initiation and control of the sessions, whilst the data transfer phase is concerned with the transfer of the speech content. The next section outlines the main features of Internet telephony signalling using two standard protocols, SIP and H.323, and one proprietary protocol used in Skype.

2.2 IP telephony signalling

Signalling is primarily responsible for enabling the communicating parties to 1) find each other, 2) establish a session, 3) agree upon session parameters, and 4) gracefully terminate the session.

In this section we will explain the basic operation of three protocols: H.323, SIP, and Skype's signalling protocol. H.323 and SIP are standardised signalling protocols, although by different bodies, and Skype's protocol is a non-standard proprietary protocol. There is a wealth of information available on H.323 and SIP in [53, 8, 101, 37]. For the Skype signalling protocol refer to [126, 131, 117, 45].

Much of the work of a signalling protocol is to maintain a consistent state within the communicating parties. Therefore a large part of a signalling protocol is devoted to ensuring correct operation of the system: which means operations that are initiated, terminate as expected. Indeed, during the protocol standardisation phase it is highly desirable to formally prove that inconsistent situations cannot arise during message exchanges. Signalling failures can manifest themselves as timeouts, looping behaviour, unacknowledged messages, and inconsistent states. Telecommunications standardisation bodies such as the ITU or ETSI have attempted to formalise the design/testing phase, whereas the IETF approach has traditionally been less formal, with validations done via early implementations and inter-operation tests.

Signalling utilises transport protocols such as TCP, UDP or SCTP. The selection may depend on the characteristics of the transport network. We will briefly discuss message loss in relation to these three transport protocols. Reliability at the transport layer is desirable, however TCP introduces signalling delay due to its three-way initial handshake. UDP on the other hand, requires only a one-way trip time to initialise an existing connection. Although this is attractive in delay terms, lost messages must be handled by the higher layers, increasing application complexity. The Stream Control Transport Protocol (SCTP) is a transport layer protocol similar to TCP,

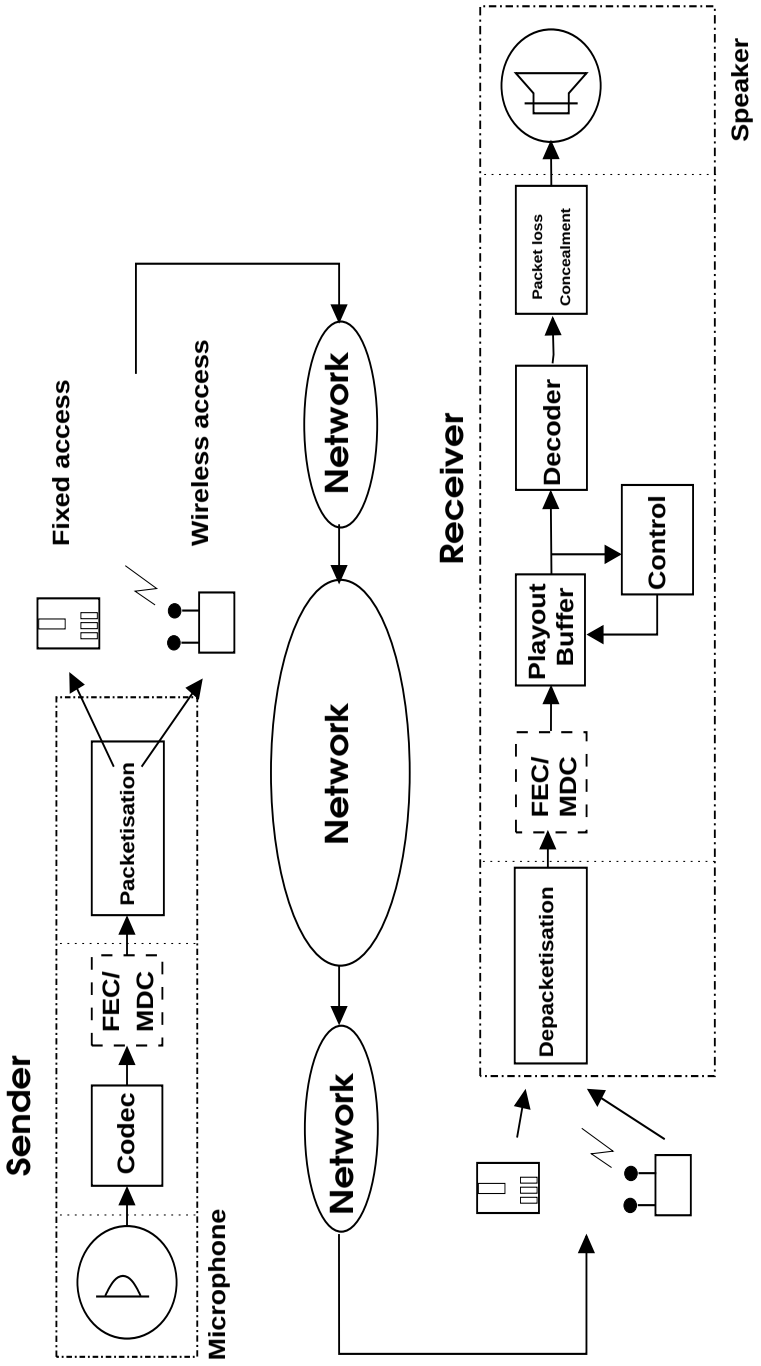

Figure 2.1: A voice journeys path

but supports complete and multiple message streams. It operates on whole messages rather than on single bytes such as TCP and UDP. SCTPs uses a 4-way handshake to initialise a session making use of a signed state cookie. This renders Denial of Service attacks more difficult to which TCP can be subject (i.e. a SYN flood attack). SCTP was originally designed to transport PSTN signalling messages over IP networks.

Two dominant signalling standards for Internet telephony have emerged during the past ten years, ITU-T's H.323 [69] and IETF's Session Initiation Protocol (SIP) [113]. The following sections will explain the basic setup operation of these two protocols plus give a short comparison of their major characteristics. Details of the particular protocol operations are however beyond the scope of this dissertation, and we refer the reader to the earlier references for further information.

2.2.1 H.323

H.323 is the result of standardisation by the International Telecommunications Union (ITU) standardisation body, the ITU-T. The ITU-T has been responsible for many standards that define operating practises within the global telecommunications industry. As the Internet has become more prevalent, H.323 has undergone a number of revisions. The current standard was approved in June 2006 (version 6). There are actually a number of separate components within H.323. It is in fact a complete protocol suite incorporating methods for both Internet and traditional telephony. More complex signalling has been necessary in H.323 to include legacy telephony.

Figure 2.2 shows an example of the signalling process between two nodes and a gatekeeper (server in the H.323 terminology). In the figure terminal A sets up a connection to terminal B. In phase I, terminal A initiates the communication to the gatekeeper, with registration, admission and signalling (RAS) messages. This part of the communication is indicated with dashed lines.

The gatekeeper provides information for A to contact B. In phase II terminal A sends a SETUP message to B on a well known signalling port. It negotiates which unit is the master and which is the slave in the pairing, establishes RTP port numbers, plus signals the logical channels. The logical channels are used for the media flows and are instantiated using a request and ACK exchange to the gatekeeper. In phase III terminal B responds with a CALL PROCEEDING message and also contacts the gatekeeper for permission to continue the call establishment. In phase IV an ALERTING message is sent from B to A via the gatekeeper once the phone is ringing at the callee. In phase V a CONNECT message is sent from B to A once the phone is answered. Both the ALERTING and CONNECT messages contain transport addresses (such as port numbers) to allow the terminals to open media channels. Phase VI uses the H.245 protocol to negotiate the codecs

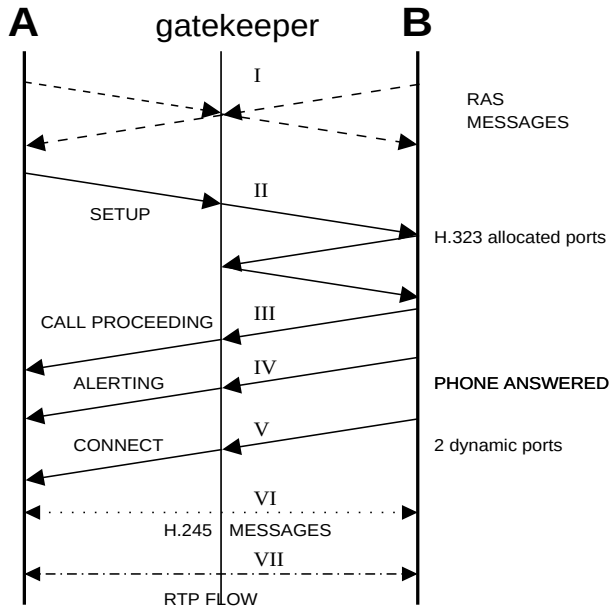

Figure 2.2: H323v3 call setup

for the session. Finally phase VII begins with the flow of the voice data. Even though this is a relatively simple H.323 setup operation, there can be a complex flow of messages. Much more material on the H.323 signalling protocol can be found in [85].

2.2.2 SIP

The Session Initiation Protocol (SIP) is a signalling protocol used in many different types of sessions. It is widely used in multimedia initialisation, but has also been adopted in presence, messaging and telecom applications. Developed within the Internet Engineering Task Force (IETF), SIP 2.0 was the first proposed standard version and is defined in RFC 2543 [51]. The protocol was further refined and published in RFC 3261 [113]. Unlike H.323 and its telephony origins, SIP is very much Internet based with its extensive use of existing IETF protocols plus an HTTP-like syntax. SIP inter-operates with external protocols such as the Session Description Protocol (SDP) for media description. SIP began life with a smaller feature set than H.323. However its adoption in other applications, notably instant messaging and 3GPP's IP multimedia system (IMS), has increased its size in recent years. For SIP material consult [18, 125]. Up to date tutorials can be found in [37, 55].

Figure 2.3 illustrates an example of a simple SIP session between two user agents, a user agent client (A) and a user agent server (B) plus a SIP proxy server. Clients are referred to as user agents in the SIP world.

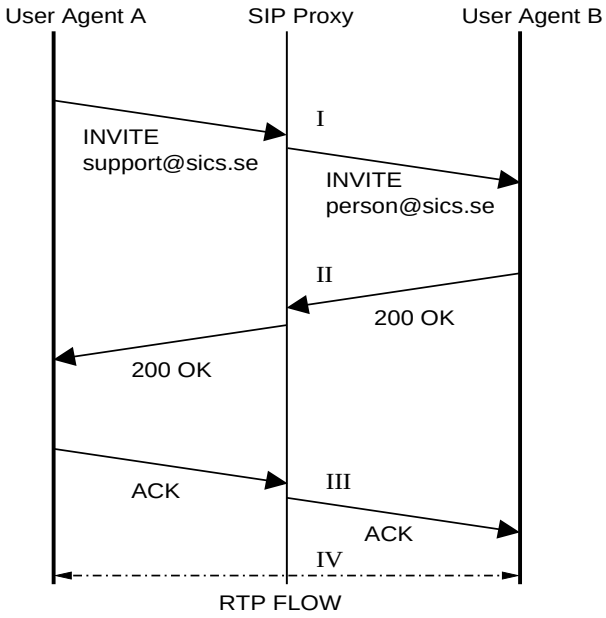

Figure 2.3: SIP setup

SIP servers play a central role as they provide inter-operation between SIP components and offer device, service and session mobility. User Agent A sends an INVITE message to a local SIP proxy. The SIP proxy then looks in a location database where "support" is registered, and an INVITE message is sent from the proxy to the user agent server B (Step I). User agent B responds with an OK message to the proxy which in turn sends back an OK to the initiator A (Step II). Within an INVITE message there are the details of the voice coding A is willing to accept, the path of the OK message must follow the same path of the original invites. The list of proxies are stored successively from sender to receiver in the original INVITE message. User agent A responds with an ACK, and if the capabilities are agreed upon (step III), the RTP media session can begin.

In both this and the H.323 case we have selected a simple scenario (A calls B). However it is clear to see the complexity difference between the two. Note, however in both cases users were assumed to be within reach of a single gatekeeper/server. In cases where redirects are needed, the number of messages needed in both protocols can increase significantly. There is a large difference between the simple call case and fully fledged telephony functionality, which has led to many add-ons and expansions to the original draft RFC.

2.2.3 A comparison of H.323 and SIP

We will now give a brief comparison of H.323 and SIP functionality. First their similarities, H.323 and SIP offer essentially the same set of services. They both provide call setup, control, and tear down. Both have basic call features such as call waiting, transfer, identification and so on. Both protocols typically rely on well known servers (or gatekeepers) for registration. Both can operate in either stateless or statefull mode and can use TCP, UDP, or SCTP for their message exchanges. A SIP user agent registers with a proxy server and H.323 terminals register with a gatekeeper; both can use IPSec or TSL. At a higher level they focus on different domains, but increasingly SIP is addressing telephony-like functionality and connectivity to the PSTN, whilst H.323 is becoming more IP compatible. This trend will probably continue from their once clear domains until the point where they cannot be easily distinguishable anymore.

We now move onto their major differences. The following discussion refers to H.323 version 6 and SIP version 2.0. SIP is under the auspicious control of the IETF whilst H.323 protocol is defined by the ITU-T. This is reflected in H.323 still being telephony based with its ISDN influence and ASN.1 coding, whilst SIP is TCP/IP based with its HTTP-like syntax. The capabilities exchange is more complex with H.323 than with SIP. The latter relies on the session description protocol SDP. SIP+SDP can issue a single request that contains most of the necessary information to initiate a session. H.323 defines its own mechanisms for such functions. SIP provides better personal mobility e.g. redirection of a callee to different locations and better support for caller preferences. H.323 has better internal developed multimedia session capabilities such as whiteboards, video, and data collaboration facilities based on the T.120 specification.

SIP is somewhat better at adding new features with its call processing language (CPL) and SIP-Common Gateway Interface (SIP-CGI). SIP also allows a third party to control a session, which is not presently possible with H.323. Due to SIP's modularity, it can more easily support a wider range of applications as we have already mentioned. H.323 has to use the ITU's H.450.1 supplementary service creation.

For Quality of Service (QoS), SIP relies more heavily on external functionality and can use any reservation protocol (COPS, OSP, RSVP) whilst H.323 recommends RSVP for bandwidth reservation, however admission control is still controlled by the gatekeeper. In terms of security SIP supports MIKEY and SRTP while H.323 relies on H.235.

Generally H.323 is more complex relying on hundreds of components such as those mentioned above. SIP initially only defined a small set of primitives (32 headers in the base specification), however in recent years, it has become rather large and the existing base standard document now extends to over 2000 pages.

As of late 2008 it is difficult to say if one protocol will become dominant, however the adoption of SIP into IMS may have an influence, depending of course on the success of IMS. Predicting the dominance of one standard over another becomes less necessary with the presence of protocol translators from Asterisk, VOCAL, and Yate which can translate H.323 and SIP messages. Additional discussions of the two protocols can be found in [28, 77] as well as those given in [68, 103, 26].

2.2.4 Non-standardised signalling

Signalling does not necessarily need to be standardised. Commercial developers find it advantageous to keep their systems propriety for monopolistic reasons, and often cite issues such as security, complexity and performance as reasons to develop closed solutions.

We will now discuss one propriety protocol for IP telephony, specifically the application layer Skype protocol. Unlike SIP and H.323 there is no centralised server/gatekeeper, there is however a central login server. Within the Skype network there are two classes of nodes: normal nodes and super nodes. We will first discuss normal nodes. Normal nodes are typically a home owner's PC and are usually behind a home firewall and/or an ISP's NAT. These nodes typically have a private IP address allocated to them. A private address is not globally routable and is defined by specific ranges which routers do forward data from. The CPU processing of normal nodes is also assumed to be somewhat limited.

Super nodes on the other hand are well connected machines and must posses a public IP address. A typical example might be a UNIX computer on a university network. Due to their connectivity and processing capabilities, super nodes perform routing and forwarding of Skype signalling messages. The load on the super node is carefully monitored so Skype message processing does not interfere with the normal operation of its host. Usually users are not unaware that the computer has been elected to super node status. The software distribution for normal and super nodes is actually identical, with different routines being invoked after initialisation. The super nodes also forward login requests on the behalf of the normal nodes, if the normal nodes cannot reach the login server.

On the first invocation of Skype, a normal node uses a pre-configured list of permanent super nodes, it then receives an update of more recent super nodes. The directory of Skype users is decentralised. Skype uses its Global Index technology to find a user with encrypted (256 bit AES) messages. In order to locate a user the procedure is as follows. A normal node sends a request to one super node, if it doesn't know itself the location of the callee. That super node then responds with four additional nodes to be queried if the person was still not found. The normal node then queries these four nodes. If the user is not found, an exchange occurs again with the same

super node. The super node then responds with eight new (and different) nodes. This is repeated several times until the user is found. Here we have assumed that the normal nodes has a public address for simplicity, in the case where it has a private address this negotiation is done by a super node on the normal node's behalf. Search results are also cached at intermediate nodes for subsequent searches.

Non-standardised solutions need to use protocol translation services if they are to inter-operate with existing solutions. Protocol translation involves taking a message from one protocol and generating a (near) equivalent message in the second protocol. We briefly mentioned some names of known translators for H.323 and SIP in the previous section. For a closed protocol the developer themself must create a translator for the desired interoperability.

There have been many publications and presentations on the Skype protocol. Prestige in being the first to reverse engineer a closed (and widely used) protocol often acts as an incentive for such efforts. Some of these can be found in [131, 117]. A more basic introduction to the operation of Skype at a somewhat higher level can be found in [126].

2.3 Firewall traversal

We will briefly look at two protocols for traversing NATS and firewalls. These are STUN (Simple Traversal of User Datagram Protocol) [114] and TURN (Traversal Using Relay NAT) [111]. Some assumptions to begin with. An external STUN server needs to have a public IP address and STUN assumes that UDP datagrams are not blocked by NATs.

The basic principle of STUN is to ascertain if a client is behind a NAT or a firewall, and what type of NAT or firewall it is behind. By requesting a reply from different servers and by requesting different ports the client can learn the bindings applied by its NAT. The Skype application has a built in STUN client, which sends a number of requests to the external STUN servers to find these bindings. STUN, however has been criticised for being unreliable and opening up security problems, a draft RFC [78] suggests that STUN is not sufficient as a complete NAT traversal mechanism.

TURN is used for a client to traverse symmetric NATs by contacting a relay NAT. A symmetric NAT works by each request from the same internal IP address/port pair to a specific destination IP address/port pair is mapped to a unique external source IP address and port. The client can use either TCP or UDP connections. TURN is normally used by clients behind a symmetric NAT that want to receive a single connection (only). It is designed so that the internal client can be on the receiving end of a connection also requested from behind a NAT. TURN is more reliable than STUN, but is more costly in terms of traffic to and from the TURN server.

This is because the server must receive and forward all the media traffic in both directions. In the case of symmetric NATs, STUN is often tried first and then TURN by clients.

2.4 Speech encoding

Human speech occupies a fundamental frequency in the range of 85-155 Hz for men and 165-255 Hz for women. Higher tones or harmonics can be heard up to 10 KHz. Encoding this full frequency range would require a sampling rate of at least twice this frequency to faithfully reproduce the speech. In a voice transmission system, the speech is sampled and then digitised according to the quality required (or restrictions) of the transmission system. In a system such as the traditional telephony system, this capacity is not sufficient to faithfully accommodate human speech's full frequency range.

2.4.1 Pulse Code Modulation (PCM)

In narrowband telephony, the frequency bandwidth is restricted to 3100 Hz, ranging from 300 to 3400 Hz. Voice in the fixed telephony system has therefore to be reduced from its original range to this 3100 Hz range (a reduction of about one third). The lower frequency of the human range is lower than that of the telephony system. This is not as problematic as it may seen, due to the perceptual system's ability to reconstruct the lower tones from the overtones. Traditional telephony does not use the low frequencies as they are very hard to reproduce with inexpensive loudspeakers.

Quantising the sampled waveform can either be done using constant steps between the sample levels or using non-constant steps, such systems are known as linear and non-linear quantisers respectively. From a 12 bit linear input signal, an 8 bit companded signal can be produced which has a similar signal to noise ratio as the original. Non-linear quantisation has the advantage that the quantisation performance is independent of the signal loudness. Its disadvantage is lower accuracy for larger amplitude signals. Two (similar) examples of non-linear quantising encodings are known as the A and μ-law companders. There are three main methods of implementing the μ-law algorithm:

- One is using an amplifier with non-linear gain to achieve companding entirely in the analogue domain.
- The second is to use an analogue to digital converter with quantisation levels that match the μ-law algorithm.
- The third is to convert the 12 bit linearly quantised representation to μ-law coding entirely in the digital domain.

In Europe A-law coding is used. The A-law algorithm provides a slightly larger dynamic range than the μ-law version at the cost of worse proportional distortion for small signals. By convention, A-law is used on an international connection if at least one country does. The G.711 standard encapsulates the A-law and the μ-law formats into a single standard [65]. G.711's simplicity (and the low SNR) makes it the default choice in the non-wireless telecommunications infrastructure.

2.4.2 Adaptive differential pulse-code modulation (ADPCM)

Differential (or delta) pulse-code modulation (DPCM) encodes the PCM values as differences between the current and a predicted value. An algorithm predicts the next sample based on previous samples, and the encoder transmits only the difference between this prediction and the actual value. If the prediction is reasonable, fewer bits can be used to represent the same information. For speech, this type of encoding reduces the number of bits required per sample by about 25% compared to PCM. Adaptive DPCM (ADPCM) is a variant of DPCM that varies the size of the quantization step to allow further reduction of the required bandwidth for a given signal-to-noise ratio. The rate of ADPCM is 32 kb/s.

2.4.3 Low bit rate models

Speech that is sampled and encoded using A or μ-law at 8000 samples per second with 8 bit resolution for each sample produces a data rate of 64 kb/s. Current speech coding techniques can produce encoded voice with rates as low as 16 kb/s which are indistinguishable in quality from 64 kb/s codec. We will discuss some of these schemes soon, however it is first necessary to explain how humans produce speech, in order to understand the technique known source filter modeling.

Human production of sounds: The lungs produce a stream of air that enters the vocal tract. The vocal tract is the pharynx, mouth, and nasal cavities. There are essentially two types of sounds: voiced and unvoiced sounds. Voiced sounds such as /a/ or /e/ are produced by the vocal chords. Unvoiced sounds have two types, the first type is fricatives such as /s/, /sh/, or /f/ which are produced when the vocal tract is constricted. The second type of unvoiced sounds are known as plosives, and include sounds such as /p/, /k/ or /t/. They are produced when the end of the vocal tract is closed, pressure is built up, and the pressure is released suddenly. There are actually additional types of sounds such as the nasal /n/ sound, but we will omit these from the following discussion.

Voiced and unvoiced segments: In order to encode and transmit speech at low bit rates, it is necessary to differentiate between the voiced and unvoiced sounds. As we will see, these sounds constitute different parts of a source filter model, and are actually transmitted separately. In order to separate them different techniques are available:

- Spectral flatness: calculated by the geometric mean of the power spectrum divided by the arithmetic mean. Unvoiced frames (typically 20 ms long) are flatter than voiced frames. The spectral flatness can also be measured within a specified sub-band of frequencies as well as across the whole frequency band.

- Energy: the square of the spectrum values of the sampled frame. Voiced frames have greater energies than unvoiced frames.

- Zero crossing points: counting the sign changes in the signal, voiced frames exhibit fewer crossing points than unvoiced frames.

Source-filter models: The most popular technique within source filter models is based on linear predictive coding (LPC). The basic idea is to model the speech generator as produced by the human vocal system, described in the previous section. The generator is a simple buzzer at the end of a tube. The space between the vocal chords (called the glottis) produces the buzz. It is characterized by its intensity and frequency (pitch). The vocal tract (the throat and the mouth) forms the tube, which is characterized by its resonances, these are known as formants.

The parametric coding process: Low bit rate coders estimate the formants, remove their effects from the speech signal, and then estimate the intensity and frequency of the remaining buzz. The process of removing the formants is called inverse filtering, and the remaining signal after the subtraction of the filtered signal is called the residue. The formants and the residue can then be transmitted to recreate the voice at the receiver. Another term for this process is vocoding, a contraction of the words voice and coding.

Decoding or synthesising the speech signal is done by reversing the process. The buzz parameters are used together with the residue to create a source signal. The formants are used to create a filter (which is the tube), and the source is run through the filter reproducing the original speech. The spectral information is well suited for vector quantisation. Compression algorithms often differ in how the residuals are treated. Typically 30 bits are used to code the 10 coefficients for basic LPC quality, and up to 18 coefficients can be used for improved fidelity.

Code excited linear prediction (CELP): In an attempt to improve on the robotic sound of early LPC schemes, a number of improvements were made that have led to methods used in modern codecs (see section 2.4.4). Multi-excitation linear predictive coding (MELPC) is based on LPC but instead of using a periodic pulse train for the voiced segments and white noise to represent the unvoiced segments, it uses mixed periodic and aperiodic pulses, a pulse dispersion filter, and spectral enhancement. The multi-pulse linear predictive coder (MPLPC) is an analysis by synthesis approach where each excitation vector consists of a number of pulses where their amplitudes that have been derived from closed loop optimisation. CELP uses an codebook (sequence) of excitation pulses as the excitation rather than the multi-pulses of MPLPC. The optimum sequence is chosen to minimise the distortion between the derived signal and the original one. At the decoder the sequence of excitation signals is passed through a long term filter and a LPC vocal tract filter to produce a block of reconstructed samples. The bitrate of CELP coders is usually in the range of 5 to 15 kb/s.

Transform coders: Transform coding tries to draw the best from waveform tracking techniques used in the PCM encoders, but also include models of the human production of speech as the source filter models do. Knowledge of the speech signal is used to select which information to discard in order to lower the bandwidth of the signal. Transform coding derives its name from frequency based techniques to code the transform coefficients in a manner suitable for voice.

Different transforms have been suggested for speech compression, we will briefly consider just two: the Karhunen-Loève transform (KLT) and the Discrete Cosine Transform (DCT). The Karhunen-Loève transform offers optimal coding performance (in terms of minimum square error) if the input samples are Gaussian distributed and the coefficients are scalar quantised. However the Karhunen-Loève transform is difficult to implement and its performance is signal dependent. The DCT is signal independent, but is sub-optimal (compared to the KLT) in that it cannot completely decorrelate the transform coefficients. The DCT is attractive since there are computationally efficient algorithms to compute it, and it retains the formant structure of the speech. The bitrate of transform coders is in the range of 10-20 kb/s, but can produce better fidelity speech.

2.4.4 Modern codecs GSM, G.729 and iLBC

GSM networks employ a LPC-based speech encoding technique called Code-Excited Linear Predictive (CELP) coding. The significant difference between CELP and LPC is that the excitation signals are not simply generated based upon a voice or unvoiced sound, but taken from stored codebooks. There are two types of codebooks, fixed and adaptive which are used in

conjunction to code the signal. ETSI's GSM has defined different rate voice codecs ranging from 6 kb/s (half-rate) to 13 kb/s (full-rate). GSM was further enhanced in the mid-1990s by the GSM-EFR codec (effective full-rate) which is a 12.2 kb/s codec that uses a full-rate GSM channel. GSM is one of the preferred speech coding schemes for wide area radio links. EFR is a fixed rate codec, however some GSM networks now use Adaptive Multi-Rate (AMR) coding [7]. AMR uses link adaptation to select from one of eight different bit rates depending on the instantaneous link conditions.

G.729 is another example of a LPC-based encoder, again a CELP codec. The coded stream consists of linear predication coefficients, the excitation codebook indices, and gain parameters. Technically it is known as variable bit rate conjugate structure algebraic code excited linear-prediction scheme (CS-ACELP). The standard rate of G.729 is 8 kb/s. It requires 10 ms input frames and produces an 80 bit output frame. It also includes a 5 ms lookahead, producing a 15 ms algorithmic delay. Annex B of the recommendation (G.729B [61]) also describes a silence compression scheme and a voice activation scheme. It also has a discontinuous transmission module, which estimates the background noise at the sender and can use a comfort noise generator at the receiver. G.729 is popular within VoIP applications, due to its low data rate and the features just mentioned. A Skype call initiated from the Internet and terminating at a PSTN connection uses G.729 for the Internet part of the path. It was developed by the University of Sherbrooke (Canada), the Nippon Telegraph and Telephone Corporation of Japan and France Telecom in 1995.

The iLBC encoder from Global IP Solutions is a block-independent orientated LPC coder [4]. Whereas LPC schemes have a memory that lead to error propagation in the case of lost packets, iLBC encodes each frame as a separate block. It therefore has a controlled response to packet loss and exhibits a robustness similar to PCM with respect to packet loss concealment [66]. The CPU resources when using iLBC are comparable to that of G.729A, but it yields higher basic quality. Although a narrow-band speech coder, iLBC uses the full 4 KHz spectrum unlike most 300-3400 Hz codecs, thus producing better fidelity. iLBC is popular in PC to PC communication and is found in tools such as Skype and GoogleTalk.

2.4.5 A (very) brief history of speech coding

The vocoder was invented in the late 1930's and is an implementation of the model of the human sound production system. Vocoders are often known as analysis-synthesis systems, where the input speech is passed through a multiband filter and each filter is passed through an envelope follower. The signals from the envelope followers are transmitted, and the decoder applies the amplitude controlled signals to corresponding filters in the synthesizer. The main motivation for this type of system was to cryptographically encode

the signals during transmission. Delta modulation appeared in 1952, it is the simplest form of differential pulse-code modulation (DPCM) where the difference between successive samples is encoded into a one bit stream. Also in the 1950s the Lincoln Laboratory at MIT conducted a study of pitch in speech detection, which led to vocoders designed to reduce the speech bandwidth. The first LPC ideas came about in 1966 from work done at NTT in Japan. In the late 1960's early real-time versions of LPC coders were implemented. The first workable LPC encoder was the US government's LPC-10 coder developed in the early 1980's [133]. The ten in LPC-10 signifies the number of coefficients it used. 1964 saw the standardisation of PCM waveform coding for fixed telecommunication networks. The implications of this choice is still with us today.

Moving forward a number of years, warped LPC was first proposed in 1980 which is a variant of LPC where the spectral representation of the system is modified. This reduces the bitrate required for a given level of perceived audio quality/intelligibility. In 1985 the Code-Excited Linear Predictive (CELP) codec was introduced [89]. The ITU's G.729 was standardised in 1996 [62]. In 1997 the Enhanced Full Rate (EFR) codec was standardised. More recently intelligent multimode terminals have appeared that can adapt their configuration to different rates, quality and robustness. These are known as adaptive multirate AMR codecs which was standardised is 1998. For an account of the early vocoder history research consult [38].

2.5 Internetworking and voice

This section deals with the networking aspects of real-time voice communication. It explains how media synchronisation is achieved at a receiver, describes formats for transporting voice, how addressing and routing effect voice streams, as well as outlining the main quality detractors in IP voice communication.

2.5.1 The Real-Time Protocol (RTP)

The RTP protocol has been developed for end to end transport of real-time media, including unicast and multicast network services. RTP can also synchronise multiple streams arriving at a single receiver. Often the RTP protocol is used with the UDP datagram service and is used in conjunction with signalling protocols we have discussed, H.323 and or. RTP was first published as a standards track document by the IETF in 1996, more recent developments have been made up to July 2003 when it became a standard [121].

The primary role of the RTP protocol with regard to voice streams is to ensure intelligible playout of the speaker's words for the listener. Without RTP, disturbances in the stream may result in incorrect playout, for

example the voice might be reproduced too fast or slow or even with parts of the sentence clipped. Recall that VoIP systems do not (normally) use synchronised clocks, therefore the timing information needs to accompany the data so that the original voice stream can be recreated at the destination. Thus far we are only discussing the operation of the synchronisation protocol, network losses only serve to compound the problem.

To recreate the spoken pattern of words and silence periods, the sending application notes where there is speech activity and when there is none. The periods of speech activity are known as talkspurts. The start and stop times of these talkspurts are recorded using media dependent timestamps into the RTP packet. The RTP timestamp is based upon the sampling instant of the first sample to be put into a data packet. The clock frequency used to derive the sampling instants is dependent on the payload media (see table 2.1). This means the system clock is not directly used, rather some function of the media rate. In the case of 8000 Hz fixed-rate PCM sampling, the clock is updated 8000 times per second (once per sampling instance). If an audio application reads blocks of 160 sampling periods (i.e. every 20 ms), then the timestamp would be increased by 160 for each packet.

In addition to the synchronisation functionality, RTP is responsible for a number of other functions such as source identification, packet sequencing, stream profiling, payload identification, and multiple source multiplexing. Out of order delivery is permitted by RTP, if the application reassembles the stream from the sequence numbers. The RTP header is shown in figure 2.4. Here PT stands for payload type and is filled in by the application.

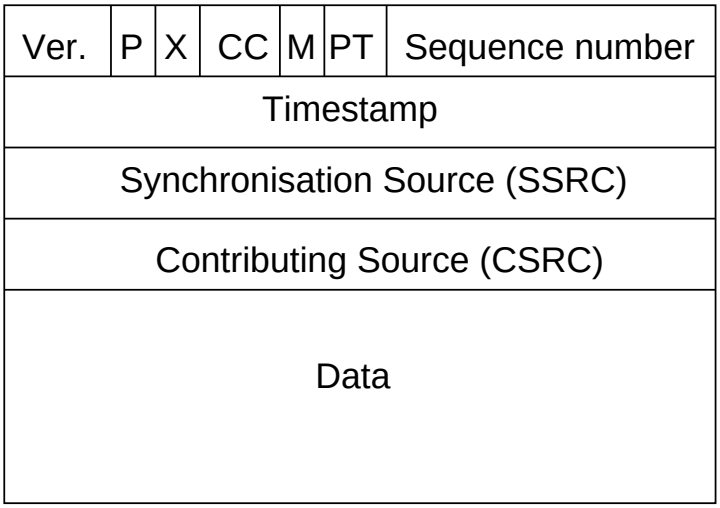

Figure 2.4: RTP header structure

A number of payload types have been specified by the IETF as shown in Table 2.1. The sequence number field is used to store the current packet

Payload type	Name	Type	Clock rate (Hz)	Audio channels	References
0	PCM-Ulaw	Audio	8000	1	RFC 3551
7	LPC	Audio	8000	1	RFC 3551
2	G.721	Audio	8000	1	RFC 3551
3	GSM	Audio	8000	1	RFC 3551
31	H.261	Video	9000	-	RFC 2032

Table 2.1: Some examples of RTP payload types

number in the stream. It is incremented by one by the sender for each data packet transmitted. The receiver uses it to calculate packet loss as well as to restore packet sequence if packets arrive out of order. A random value for the sequence number is selected at the start of a session in order to make DoS attacks on the session more difficult. The synchronization source (SSRC) identifies the synchronization source. This value should be unique and is also chosen randomly, with the intent that no two pair of synchronization sources within the same RTP session will have the same SSRC. The contributing source (CSRC) identifies the contributing sources for the payload contained in the packet. The CC field indicates the number of CSRC identifiers.

Refinements to the RTP protocol have primarily focused on header compression. The purpose is to reduce the combined size of the IP, UDP and RTP headers. In 802.11 networks, headers can be significantly larger than the payload itself. This is partly due to the large 802.11 frame headers. One proposal has been the Robust Header Compression (ROHC) scheme specified in RFC 3095 [132, 80]. The scheme is called robust as it can deal with relatively high error rates. Note that ROHC and similar schemes do not compress the payload, only the headers. The typical compression rate is from 40 bytes down to 4 bytes. Also note that shorter headers reduce the possibilities of bit errors in the frame, since they constitute fewer bits in the air.

In header compression schemes, a compressor and decompressor exist before and after the link where compression is needed. The basic idea is to send a complete header at the start of a session, and from then on only updates to this complete frame, called delta frames. Often more than one identical delta frame is sent to allow for low numbers of losses. There are different states within the compressor, such as full state, first order state where the static fields have been detected, second order state where dynamic fields are suppressed and replaced by logical sequences, partial checksums so the receiver can predicts and generate the next sequence number and so on.

The Real Time Control Protocol (RTCP) is a companion protocol to RTP. RTCP is used in one-to-one or multi-party sessions by receivers to

inform the sender of the stream quality they are receiving. Observed packet loss, delay, and jitter are fed back to the sender. RTCP can generate data based on the start of the session or from the last report arrived. To calculate the round trip delay, the sender transmits a report containing the time the report was sent. On reception of this report the receiver records its current time. Therefore two times are now recorded within the report. When transmitting the report back to the sender, the receiver subtracts the time it held the report from the time it initially put in the report, therefore accounting for the time it held the report. Using this information, the sender can calculate the round-trip delay and discount the time spent processing the reports at the end points. This can be done in both directions if asymmetry problems are suspected.

Further extensions have also been proposed to RTCP [22, 102, 35]. These basically extended the information put into the reports to include end systems artifacts such as buffer levels and estimations of the quality received.

2.5.2 Addressing, routing, and timing constraints

The end-to-end delivery of voice packets is the joint responsibility of the networking and terminal equipment. This includes the end systems, access points, layer 2 switches, firewalls, NATS, IP routers and interchange points (iX's). In an MPLS network one has the label switch routers at the center of the network and the label edge routers at the extreme points of the network. To some degree, all network elements affect the end-to-end transmission of voice data, in particular delay. It has been argued for some time within the Internet community that several classes of traffic (including telephony) deserve higher priority than other data in order to reduce delay within equipment that queues or stores packets [3, 15, 9].

Simplistically the destination IP address is used to route each packet toward the target terminal. The UDP port field is used to demultiplex the data at the receiver to the correct application. The source identification field (SSRC) is used to locate the correct RTP flow within a session. For the actual routing of IP voice data, the normal IP routing mechanisms apply. Path information from a company, home, or university network is provided to the backbone using interior link-state routing protocols such as IS-IS or OSPF. In the backbone network, routes are determined by peer agreements and the inter-domain routing protocol BGP. In a MPLS network voice traffic may be given its own path through the label switched path if the particular operator has sufficient traffic for it to be worthwhile.

It is possible that a router may send some packets of a single stream via one route and other packets via another route. A more likely event however is that the route between two parties may differ over longer time intervals. That is to say, there is less path stability over longer durations. Route changes are an inherent fact of the IP infrastructure. However the issue

for voice traffic is that the delay requirements are not exceeded. A typical route from Europe to the US consists of approximately 20 intermediate routers. Routes are not necessarily symmetric, which means that the number of traversed routers is not the same in each direction. The end to end implications for voice depends on the traffic on each link and router, not purely on the number of hops.

Also it is possible that some of the voice packets will be lost, due to congestion in the routers, discarding algorithms such as RED, or link problems. Again, loss is unwelcome in telephony-like applications. Correlated losses are more likely to cause problems for the voice receiver which are more prevalent in wireless networks. Additionally non-licensed spectrum technologies are more prone to disturbances and losses of frames [16]. However, 802.11 provides a link layer retransmission protection that can alleviate frame loss on wireless access links to some degree at the expense of a little delay. Other sources of problems for IP-based voice are heavy traffic loads on shared links, poorly dimensioned links, long-delay link technologies (e.g. satellite links) and misconfigured equipment.

Over-provisiong and priority schemes can make acceptable quality IP telephony sessions possible. Lost packets cannot be retransmitted due to the overall delay budget for conversations, and therefore protection in the form of redundancy can be introduced at the sender, and concealment at the receiver can lessen the audibility of losses by interpolating small gaps in the sample sequence.

2.5.3 Packet delay

The *network delay* is the time taken for a packet from the operating system boundary at the sender to the operating system boundary at the receiver. The operating system boundary is usually thought of as the interface between the user/supervisor modes. In UNIX this would be user/kernel boundary. The *end system* delay varies widely from operating system to operating system and between VoIP applications. The delay incurred by an end system can vary from 20 ms up to 1000 ms, irrespective of the stream characteristics [47].

Since real-time voice has constraints on the end to end delay for the samples to reach the listener, we will now consider the constituents of the delay. From a routing perspective the path with the lowest delay is desired. This implies a propagation delay based upon distance. In reality finding the length of a link is not trivial, as the links can traverse non-obvious paths, be split into different paths and so on. This delay constitutes the *deterministic delay*, even if it is non-trivial to obtain. There are processing and queuing delays along the path too. Each packet needs to be processed by several routers. In most cases this means looking at the IP address within the header and finding the correct interface to forward the whole packet to.

Deciding upon which interface to select depends on matching the IP address in the header with a routing table. The path with the longest matching prefix is chosen. Whilst forwarding or processing, the packets behind it must wait, causing random delays. The instantaneous queuing delay at a router depends on: the traffic arriving at that instant, the processing rate of the router, the length of the packet and the number and lengths of packets waiting ahead of it. Due to processing and queuing delays, the original voice packet stream becomes distorted requiring resynchronisation at the receiver. The processing time for a voice packet is generally constant, but the queuing delay is variable, as it depends on the factors just mentioned.

Measuring one-way delays is not trivial without synchronised clocks [93]. One-way delays may be important from an operators perspective, but cannot be heard or distinguished by the users. Therefore it is easier to consider the round trip times. This is because most spoken words are responded to, creating feedback in the speech pattern between the two (or more) people. Only when the response is heard can a speaker have some idea of the delay. The tolerable round trip delay is typically in the order of 400 ms. Therefore the processing and queuing time per router should not exceed 10 ms if there are 40 router hops in the end to end path (20 in each direction). The interactivity of the conversation is affected by the round trip time, however defining an interactivity metric is not that simple, due to the human ability to adapt to varying delays. Conducting tests with pairs of people is more demanding than with individuals.

We have measured the network delay using the RTCP protocol, which is part of the RTP standard [121]. Because the sender and receiver exchange time reports it is possible to calculate the networking delay, by subtracting the time reports were held at the end host. Since these reports are exchanged every few seconds, the delay variations can also be found. This can be done in both directions to see if any significant asymmetries exist.

2.5.4 Packet jitter

Packet jitter is simply the variation in the delay. If isochronously sent packets arrive at the receiver with differing delays, the end to end transfer has introduced jitter into the voice stream. Jitter can have undesirable effects in a system. In voice systems it can lead to lengthened delays, due to the need to capture late packets. Loss can be incurred if the packet jitter is greater than the receiver buffer at that instant. One positive aspect of having a buffer in a voice system, is that it allows for a tradeoff of loss against delay. This means that the system is tunable to some degree, by using a buffer length that induces loss and reduces delay or increases delay and decreases loss. In a voice system the loss/delay balance should be based upon the acceptable round trip delay and the acceptable loss rate of the coding scheme.

Voice jitter is compensated for by re-aligning the timing of the packets to their recorded times. The jitter definition by the IETF is stated to be the mean deviation (smoothed absolute value) of the difference in packet spacing at the receiver compared to the sender for a pair of packets [121]. This is shown in equation 2.1.

$$J_i = J_{i-1} + \frac{(|D_{i-1} - D_i| - J_{i-1})}{16} \tag{2.1}$$

J_i is the current jitter value
J_{i-1} the previous jitter value
D_i is the current delay between two successive packets
D_{i-1} is the previous delay between two successive packets
16 is the smoothing constant

The jitter units are the timestamps used in the RTP packets, which is typically the packetisation interval multiplied by the sampling rate. If S_i is the RTP timestamp from packet i, and R_i is the time of arrival in RTP timestamp units for packet i, then for two packets i and j, $D_{(i,j)}$ (where j is sent after i) may be expressed as:

$$D_{(i,j)} = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \tag{2.2}$$

$D_{(i,j)}$ Delay for packet pair (i, j)
R_i Reception time for packet i
R_j Reception time for packet j
S_i Send time for packet i
S_j Send time for packet j

In practice one can calculate the jitter as the difference in the relative transit time for two packets. This is because the S_i and S_j are sent at (roughly) constant intervals, i.e. the time difference between two successive packets in RTP timestamp units. The jitter value is sampled and sent in RTCP reports, so that the sender has a quantitative notion of the packet delay variability in successive reporting intervals.

One other measure closely associated with jitter is the difference in the inter-arrival times. This is simply the diffence between the arrival times of two consecutive packets. Given the packetisation time it is simple to calculate by how much the packet separation has been distorted. One other method for measuring the separation is to consider the difference in the time between when the packet arrived and when it *should* have arrived. Note that this measure can be either positive or negative.

2.5.5 Packet loss and redundancy schemes

Packet loss is the major quality detractor in Internet telephony as far as the network is concerned. Packet loss implies lost speech frames. Packets from a stream can be discarded from router queues, either due to buffer space restrictions or by explicit congestion alleviation algorithms. Some algorithms implement a random mechanism for discarding packets in order to ensure fairness between flows. Under-dimensioned as well as poorly administered networks often yield higher loss characteristics.

Transmission errors on fixed modern networks are rare, while frame losses are still prevalent for wireless networks. In wireless networks the interference from competing transmissions and weak signal conditions are the main causes of frame loss. Switching between base stations or access points also leads to bursts of lost packets. Wireless networks usually implement mechanisms for link layer retransmissions; nevertheless conditions may arise that lead to IP packet loss once a number of retransmissions of a frame has been unsuccessfully attempted.

Depending on where in the phrase the losses occur, relatively large differences in the intelligibility can be perceived [56]. The speech may be encoded to make it more resilient to loss. Redundancy optionally adds delay to the system as additional packets need to be received if the receiver is to recreate lost packets. The delay incurred depends on the lost packet's position in the redundancy block. The size of the block should be chosen so as optimise a delay-quality tradeoff function as in [112].

Sample data from lost packets can to some degree be masked by speech codec-specific algorithms. That is, lost speech frames can be masked by the speech decoder where gaps are detected. Frames are created from those frames that are present, usually the ones just before and after the missing frames. Recreating lost frames is desirable as replying any sound has been shown to be perceptually more tolerable than just silence. Understanding the impact of losses in perceptual terms is not a trivial task. PESQ is one solution to assessing the influence of loss on a phrase, another is subjective user tests. A controlled response to packet loss is desirable from a speech coding point of view, thus this has been one goal of iLBC, G.729 and GSM. Studies by researchers in the 1990's advocated the use of forward error correction since losses were correlated but often only by a small amount [10]. Forward error correction (FEC) and multiple description coding (MDC) are techniques to reduce the probability of gaps in the decoder input. In IP networks FEC and MDC redundancy packets are sent time shifted from their originals. In voice communication the receiver buffer algorithm needs to make a delay calculation how long to wait for the redundant copies, assuming the originals were lost. More sophisticated scheme can feed back this information to the sender to regulate the amount of redundancy. This can be done in Reed-Solomon FEC coding for example. A comparison of the

rate distortion for these techniques can be found in [82]. Other techniques for loss analysis are [74, 73, 75]

The widespread deployment of local wireless access has changed this view somewhat. This is because the loss pattern in this setting is quite different than the fixed wired Internet (where losses were not correlated to the same degree). Unfortunately, in wireless systems correlated losses are more common, primarily due to poor signal reception at the receiver.

In practice, packet losses can be detected using the sequence numbers in the RTP header and loss ratios over time can be reported using the RTCP protocol. Further quality extensions have been proposed, such as sending back the loss distribution or finite-state model parameters (such as the Gilbert 2-state model) of the observed loss pattern. More meaningful information by the receiver (or indeed an access point) can lead to better solutions, whether it be over the last link or end to end.

Chapter 3

VoIP quality aspects

This chapter is divided into two parts, quantifying quality and some standardised approaches for calculating it. As tools and methods have been developed for the telephony industry, it seems natural to re-use them for Internet telephony where appropriate. We will introduce two standardised methods for estimating VoIP quality as they are used within this dissertation. For a more in-depth treatment of objective and subjective methods consult [106].

3.1 Quantifying quality

Although most people have a good feeling of what good quality (or more accurately fidelity) means during electronic communication, it is not straightforward to translate this into measurable parameters of a system. First the system we are dealing with is a distributed system and each component has its own individual attributes. Second people are involved in the assessments, and add inevitable human variations. Third, people are adaptable, therefore ratings tend to change over time and finally the situations differ from environment to environment.

The simplest form of quality rating for speech would be something descriptive, for example 'EXCELLENT' for a speech sequence that was almost glitch-free down to 'POOR which was barely understandable. Different words could be used, or any number of intervals between the extremes choices, however studies have shown, in a descriptive setting, three intimidatory steps are reasonable. Numerically, it is somewhat easier to get a finer scale, however more than ten intervals often leads to fuzziness between the intervals.

3.2 Measuring quality

Determining an accurate quantitative measure for human speech fidelity is desirable, but impossible. The best one can achieve is a qualitative rating that has been established in a rigorous and controlled manner. Typically test listeners and controlled auditory conditions are used for people to rate speech coder performance for example. It can be expensive and time-consuming. There are tools and methods that map qualitative assessments to quantitative values, however they will always be, to some degree, approximate. If one can show however, that there is reasonable correlation between the qualitative and quantitative results, and under what conditions the correlation holds, then this solution may be acceptable to some users. Some objective tools, such as those which use signal processing techniques, have shown this correlation and hence have found acceptance within the community. Therefore with some degree of confidence, the software developers can justify their techniques have proven success and give results as real people would.

3.3 Quality tolerances

When human speech is uttered, the time taken from when the pressure waves leave the mouth to the sensation of hearing is a fraction of a second for a nearby speaker. We have evolved to expect, and actually need, to hear our own voice. This is in order to be sure that we are saying what we really want to. The development of the human speech and hearing recognition has however taken place via face to face meetings. Thus, extra visual or body cues are available when uncertainty is present. An example of such 'understanding' is when a language is being spoken that we do not understand. We can sometimes guess the meaning from gestures, facial expressions and intonation.

On the other hand, impaired speech requires extra concentration from the listener, that is we are not used to processing distorted or missing segments, visual and auditory clues are more difficult to interpret. Somewhat similarly is communicating with people from afar, we don't receive the original speech samples and visual cues are harder to see.

In IP voice communication systems the visual cues are not existent, thus making intelligibility more important. In order to hear one's own voice a very short delay is introduced between capturing the recorded voice and replying it for the speaker. This is particularly applicable when using headsets. The introduced delay is in the order of 5 ms.

As far as the delay in the system is concerned, it is obviously desirable to keep it below some maximum. This is in the order of half a second. Delay is discussed from a networking perspective below. Recent results have shown that delay is not as significant as once postulated, at least in VoIP systems.

Traditional telephony standards have been much stricter with respect to delay budgets [134]. If one is not in a highly interactive conversation, then higher delays can be tolerated than those suggested by telecommunication standards. This is particularly true in situations where people use computers, delays are expected by users (operating system hiccups) and therefore their delay expectations also become relaxed from the communication system.

If users are engaged in quick voice exchanges, delays will frustrate their conversational style. Therefore, introducing the factor of interactivity into an objective quality measure is still under research. The following studies have looked at conversational interactivity [136, 46, 48, 107, 49]. The last reference in this list proposes the potential impact of interactivity on the perceived quality for Internet telephony services.

Where delays and losses are experienced at the same time, it has been shown that the influence of losses is much more significant with respect to the perception of quality degradation than the influence of delay. This implies that people are able to make a transition from highly interactive scenarios to a more measured communication style. In fact this transition appears to be somewhat bilinear, that is, the quality degradation from an interactive mode to a simplex conversation mode occurs in two linear steps, with the break at about 400 ms. Varying delays can be disturbing, due to the listener not being allowed to settle into a single mode of operation. For more information on the influence of delay on Internet telephony see [14].

3.4 Quality and noise

The quality of voice communication actually depends on many (independent) factors. The effect of noise, be it in the electrical circuitry, or in the surrounding environment can be a determining factor in the perceived quality.

The quality of the components is a key issue in voice systems. Lower quality components can leave voice sounding thin i.e. a lack of bass in the speech. Background noise, caused by poor grounding or shielding of the analogue components is frequently experienced as low frequency humming in the system. Internet telephony systems that use on-board sound cards can introduce noise of this nature into the signal. USB headsets are helpful, and they also alleviate the need for echo suppression.

The environment is another factor, whether a noise source is remote (distant from) or local (close by) to the speaker. In the remote case, the non-speech parts of the voice should be suppressed so as not to interfere with the spectral analysis of the voice processing. Undesirable noises from similar frequencies and volumes will be encoded into the signal, sent, and reproduced for the listener. Often listening to a remote speaker in the pres-

ence of background noise is more difficult than when background noise is present locally.

Research in the signal processing field has studied the issue of noise in systems [115]. Important speech parameters such as the intelligibility, clearness, or naturalness of speech can be improved by signal processing using digital, analog, or hybrid solutions. A robust, low complexity, speech enhancement algorithm has been proposed to show the advantages of a purely digital, purely analog, and a hybrid digital-analog implementation in [116].

In terms of testing systems with controlled noise, the ITU conducts tests with standardised background noises. These are known as mean noise reference units (MNRU) [63]. Typically well defined noise patterns of fixed modulated noise are presented at the beginning of each test. Each sample represents an example distortion corresponding to a five grade impairment scale (excellent to poor). The MNRU has been used extensively in subjective performance evaluations of conventional telephone and wide-band voice systems.

3.5 The ITU-T E-model

The E-model is intended as an off-line planning tool. Due to its simple form it has found applications into on-line assessments as well. Network planners can input parameters from a system and obtain a numerical value (between 1 and 100) representing an estimate of the perceived quality. One important point of the E-model is that loss, delay, jitter, speech coding and echo parameters are combined *linearly* to calculate the so called impairments that result in the score. The E-model assumes the parameters are *independent*. Another important (selling) point of the E-model is that the numerical scores correlate well with subjective tests, indicating that this estimation is indeed possible. Since the linear combination is simple, and most of the parameters are easily measurable, the E-model has been popular for a number of years.

The E-model also indicates how network impairments and speech coding can be combined to give an approximate estimate of voice quality. It is important to state that there are many tunable parameters included in the model, 19 in fact, not including the different speech encodings and loss concealment methods. Interestingly, jitter is not explicitly included as an input parameter. As jitter can affect whether packets arrive in time for playout or not, late packets for a real-time audio application are akin to network loss or delay, which are included in the model.

Table 3.1 shows scalar values known as the R-value derived from the computational model. They are relatively consistent with subjective scores, i.e. real user estimations of the speech quality, shown by their respective mean opinion scores (MOS). Mean opinion scores are derived by replaying samples to a naïve set of listeners who rank the quality on a scale from 5

User satisfaction	R-value	MOS score
Very satisfied	90	4.3
Satisfied	80	4.0
Some users dissatisfied	70	3.6
Many users dissatisfied	60	3.1
Nearly all users dissatisfied	50	2.6

Table 3.1: The ITU's E-model and MOS scores

(best) to 1 (worst). The R-value is defined as shown in equation 3.1.

$$R = R_o - I_s - I_d - I_{e-eff} + A \tag{3.1}$$

R = rating value
R_o = signal to noise ratio (noise sources)
I_s = voice impairments to the signal (side-tones and quantisation distortion)
I_d = delay and equipment impairments
I_{e-eff} = packet loss impairment (including random packet losses)
A = advantage factor (compensation of 'other' factors)

Each of the factors is calculated and subtracted from the maximum of 100 to obtain the R-value. The impairment due to the delay is denoted by I_d. Two different values are defined, $I_d = 0$ if the absolute delay (T_a) is less than 100 ms, i.e. no impairment or an increasing I_d if the delay is over 100 ms. A number of amendments have been to incorporate non-random losses into the model [1, 27]. The effect of packet loss on the R-value is given by the I_{e-eff} term. The I_{e-eff} is defined in the E-model as:

$$I_{e-eff} = I_e + (95 - I_e) \cdot \frac{P_{pl}}{P_{pl} + B_{pl}} \tag{3.2}$$

P_{pl} = packet loss probability
B_{pl} = packet loss robustness

For G.711, $I_e = 0$. This means for situations without loss, G.711 provides the best speech quality. The advantage factor A, is a value that indicates how tolerant users can be when using telecommunication equipment. It can be seen as a willingness to trade quality for operational convenience. One example is with mobile telephony, where users accept lower quality since they have the luxury of being mobile. One other example could be an advantage factor, as mentioned, where higher delays are tolerated when using a computer as a communicating device rather than a telephone.

3.6 Perceptual Evaluation of Speech Quality (PESQ)

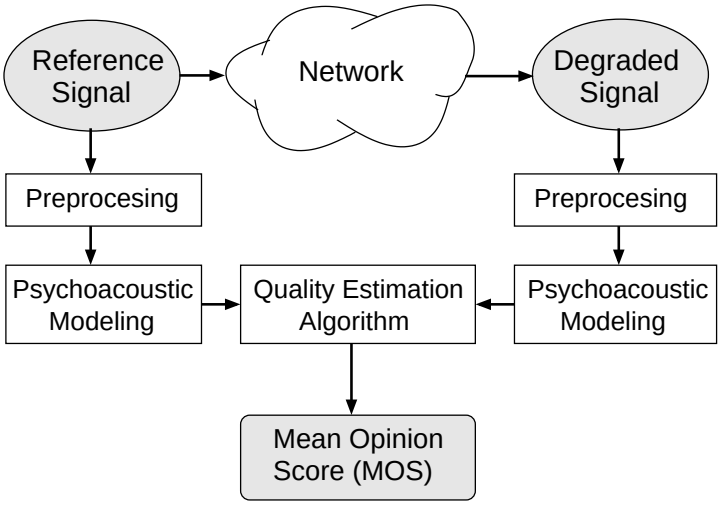

Figure 3.1: The PESQ processing structure

PESQ MOS	Linguistic equivalent	Quality degradation
4.5	Excellent	None
4	Good	
3.5	Good/Fair	Moderate
3	Fair	
2.5	Fair/Poor	Severe
2	Poor	
1	Bad	

Figure 3.2: A quality degradation scale

Although the E-model is popular for estimating quality using network parameters, it has shortcomings. As we have seen, the bursty effects of packet loss on speech quality are not well addressed in the E-model. A later development by the ITU was to develop a scheme that could improve on the E-model by estimating the impact of speech coding and losses on the original speech signal itself. The solution, the "Perceptual Evaluation of Speech Quality" or more commonly known PESQ, addresses these issues [135].

The idea is to estimate the *degradation* of the coding and loss on a speech sample using a model of the human auditory system. Figure 3.1 shows the functional units of PESQ. A reference speech signal is transmitted through a network that results in a quality degradation corresponding to the coding

used and the network losses. PESQ analyses *both* the reference and degraded signal and calculates their representation in the perceptual domain based on a psychoacoustic model. The disturbance between the original and the degraded speech signals is calculated by a quality estimation algorithm and a corresponding subjective mean opinion score (MOS) is derived. The evaluation of speech quality using PESQ is performed off-line due to its computational complexity. If one assumes a 20 ms packetisation and an eight second sample, the sequence would then be 400 packets. As an indication of the time needed to compute a PESQ score, a sequence with ten losses requires approximately two seconds of processing time for G.711 coded speech on a Pentium III computer. G.711 yields the maximum PESQ score (4.5) in the absence of loss, however it is particularly sensitive to packet loss even when concealment is used.

PESQ's validity has been shown by its ratings being sufficiently correlated to subjective ratings as we discussed in the introduction of this chapter. More recent research that correlates PESQ with subjective scores, shows that some small transformations are needed to better align PESQ to MOS [108].

3.7 Other measures

Recent work by Hoene et al. proposes a real-time implementation of PESQ called PESQlite [54]. The idea is that using PESQ in real-time is too slow for real-time use. Hence PESQlite reduces the complexity by making simplifications to the PESQ algorithm, e.g. by using constant length test samples and non-time alignment of the degraded samples. PESQlite is currently only available for G.711 coding.

One other alternative for an objective measure is to use machine speech recognition as a MOS predictor [76]. The technique uses a word recognition ratio metric to reliably predict perceived quality. This ratio is speaker-independent, whereas the absolute word recognition ratio of a speech recogniser is speaker dependent. The relative word recognition ratio is obtained by dividing the absolute word recognition ratio with the value at 0% loss. The results show that human and machine based recognition techniques are correlated, although not linearly. It is also been found that human-based word recognition ratio does not degrade linearly once packet loss exceeds 10%, due to performance limits of the codec.

Chapter 4

Packet-switched voice research: A brief history

In this chapter we provide a brief history on the development of Internet telephony. The focus is primarily on research related activities, rather than commercial ones. We will however mention standardisation efforts, as they are significant in the history of Internet telephony.

4.1 Pre-Internet days (1970-1980)

Well before the modern Internet was devised, people where investigating alternatives to the traditional telephony system for carrying voice. The earliest accounts of packet switched networks can be found in the signal processing community. Researchers and engineers were looking for computationally efficient methods of compressing voice for transmission over low bandwidth links. In fact, advances in low data rate coders and the deployment of a distributed packet switched network led to some of the earliest findings [91]. The details of the networking are often omitted, but the idea was to block-code voice for transmission. Much of the focus was on LPC and entropy methods. Blankenship et. al described the Lincoln Laboratory digital voice terminal system in a technical note published in 1975. Accounts of the early days of vocoder work can be found in [42] and the small amount of networking in [25].

In 1973, the Network Voice Protocol (NVP) was developed by Danny Cohen, then at of the University of Southern California. NVP was used to send speech between distributed sites on the ARPANET using LPC coding. The protocol was implemented in two parts: a control protocol and a data transport protocol. The control protocol is the equivalent of today's signalling solutions (H.323/SIP), and the data transport akin to packet transport using pure UDP without RTP.

William Naylor in 1974 published "A status report on the real-time

speech transmission work at UCLA" [97]. In this work he expresses the need to smooth out the variable delays of a stream of packets (speech, for example) on packet switched networks. "This is to preserve the continuity of the stream".

In 1977, Cohen wrote that the packetisation algorithm and data rate should be varied according to the network load [24]. Interestingly, this adaptive approach of reacting to the network load become popular many years later. Cohen also states that the time spent at the receiver (called the 'waiting period' in his paper) should be a function of the network performance. Indeed, Cohen states that the parameters in a real-time voice communication system are heavily dependent on the network performance, and a systematic method of predicting it must be developed. In the same year (1977) Naylor published his doctoral dissertation "Stream traffic communication in packet switched networks"[98].

James Forgie published "Speech communications in packet-switched networks" in [34]. In the first half of [38], Gold gives a background of speech coding techniques available in 1977. In the second half of the paper he gives an explanation of packet speech experiments performed across the ARPANET. He considered the delays both in coding and the transmission of different sized packets, as well as the variation in the delay. His conclusions were that reassembly needs to be done at the receiver via buffering, however vocoder techniques could be used without significant loss in the speech quality.

Among the other early works in this period was John Gruber's "Variable delays in a shared network environment handling voice traffic" in 1979 [43]. His vision was a hybrid packet and circuit switched network called 'Transparent message switching' for handling both voice and data traffic. The ideas were novel, preliminary, and pre-cursory to both ATM and today's multiservice Internet. The basic entities processed are messages rather than calls. The messages belong to an established call, however they may be completed or blocked at the network periphery. Voice messages are given priority when delays are excessive, however when loss is being experienced, voice messages could be discarded. This observation is interesting in that, delay was seen as a more critical issue than loss in those days.

In December 1984, Warren Montgomery published "Techniques for Packet Voice Synchronization" in [94]. He considers the local and wide area network scenarios separately to synchronise VoIP receivers. The paper discusses four types of delay calculations: blind delay as the worst-case assumption, round-trip measurement as estimated by the sender, absolute timing using a master clock, and accumulated variable delay using a time stamp as synchronization methods for packet playout at a receiver. Round trip estimates are sufficient for the local area case, whilst more sophisticated methods are needed for the wide area case. He suggests that the addition of timing information and incorporating extra delay at the receiver should be sufficient to

yield satisfactory voice quality in the wide area case. This is the approach taken by most modern real-time packet voice applications, as it is effective, relatively simple, and cheap to implement.

4.2 A decade of research (1980-1990)

The early eighties produced a flurry of packet-based voice research. Probably fuelled by other developments in IP research. This period is sometimes referred to as the golden age of IP networking research.

Much of the voice focus was on solutions, mostly theoretical, for buffer design and sizing [6, 5]. Work by Naylor and Kleinrock described general design methodologies for the design of jitter absorbing buffers [99]. Some early performance evaluation papers were also published, being both theoretical [128] and simulation studies [130]. Mackie et. al even considered a complete system [90]. Weinstein [140] and Adam [2] gave accounts of experiments using the ARPANET and the Cambridge ring LAN respectively. These relatively early works gave some valuable insights into the issues we face today.

A loss concealment scheme was published in Jayant and Christensen's 1981 article "Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure" [72], which was a form of Multiple Description Coding (MDC) using separate descriptions of the speech signal. Psychological and quality aspects were also beginning to emerge, Goodwin authored a book about the interaction of 'speakers and hearers' in the early 1980s [39].

Some researchers looked at quality aspects particularly for packet-voiced systems [44, 83]. Holtzman looked at the interaction between queuing and voice quality in variable bitrate packet voice systems [57]. Network delays [40] and statistical multiplexing of voice [86, 127] also appeared for packet voice, along with some early priority schemes for voice traffic [96]. The ITU released numbers of important specifications [59, 65, 58, 60]. Importantly in this decade, IP and ATM were competing technologies, with ATM keeping voice foremost in its multiservice solution. Basically the ATM Forum proposed five different circuit emulation services, depending on the capacities required. Although both IP and ATM were technically viable for both voice and data, the flexible data transport structure of IP, plus the development of the HTTP protocol, and lower hardware costs effectively sealed the fate of IP over ATM.

4.3 Emergence of telephony applications (1990-1995)

In the early nineties, Domenico Ferrari's group at the University of California at Berkeley published a number of significant papers about the effect of jitter

and delay on real-time communication applications [138, 32]. Their work proposed a distributed mechanism for controlling the delay jitter in a packet-switching network. They argued that if the advantages were sufficiently high, then the implementation was worthwhile. Although no such scheme was deployed, their work is still widely referenced as seminal.

Events such as IETF meetings and the space shuttle missions helped popularise conferencing over the Internet [21]. The space shuttle sessions were reception only, whereas people actively participated in the IETF meetings. Both showed that the Internet could support sessions of thousands of people, both passively and actively using IP multicast. The impact factor was significant, however it was only really realised by the Internet community at that time. It was the first time voice and video could be seen by normal users, via mechanisms other than radio or television. Additionally the MBONE sessions permitted group participation. Research continued on IP multicast, although it never really caught on for large scale deployments. A suite of real-time applications were produced, notably VIC, VAT, and wb (whiteboard) from the Network Research Group at LBL, USA [71] and at GMD, in Germany with Nevot [119].

Between 1993-1996 Jean Bolot wrote a series of papers that reported on, and characterised the loss and delay of audio packets on the Internet [10, 11, 12]. They were largely theoretical studies supported by experimental evidence that advocated the use of techniques such as redundancy protection against packet loss. In the late 1990s, a tool called Freephone was developed by the Rodeo group at INRIA in Sophia Antipolis, France which implemented FEC mechanisms [109]. At that time all the applications were UNIX based, as this was the only (open) operating system for Internet applications.

These earlier works led to a standardised transmission protocol, RTP for use with real-time media flows. One of the authors was Van Jacobson, who gave a Sigcomm tutorial in London 1994 entitled "Multimedia conferencing on the Internet" [70]. In this presentation he suggested using a simple synchronisation protocol to restore the original timing information at the receiver and a small adaptable buffer to absorb delay variations. Although the idea had been suggested by others previously, the presentation was influential and moulded the approach taken by researchers for many years. It also promoted the development of the RTP standard.

Henning Schulzrinne's 1993 PhD dissertation "Reducing and characterizing packet loss for high-speed computer networks with real-time services" studied congestion control, scheduling, and loss correlation for real-time traffic [120]. Schulzrinne highlighted the practical importance of scheduling packet audio within the operating system. He was also one of the main contributors to the RTP protocol and has produced the most seminal research (over 50 publications) and prototypes within voice research, including SIP and RTSP.

Tools such as the Robust Audio Tool (RAT) came from UCL in London in 1995 [52]. RAT, with its simple redundancy scheme, sending one compressed version of the packet in the following one, was an simple example of utilising redundancy. RAT was intended for both group, and one-to-one conferencing. Somewhat surprisingly, RAT and VIC were still being maintained today as part of the AVATS project (formerly SUMOVER) at UCL, London.

4.4 Early deployment days (1995-2000)

Internet telephony seemed to succeed as a business, therefore many researchers took to looking at the core issues again. Some of the important VoIP papers appeared in 1995. The problems of packet loss was addressed in [12, 52, 118]. Packet jitter and playout were readdressed in [129, 95, 124]. Some fundamental design issues for the Internet were proposed in [123], and a book on speech coding and synthesis was published by Kleijn [84] (one of the creators of the iLBC codec). Also one of the first papers on IEEE 802.11 and VoIP was published in the same year [139].

In 1994, an IETF developed QoS mechanism arrived, called Integrated Services [15], it influenced many researchers and their real-time media agendas. Arguably, the proposal of new QoS mechanisms stifled pure packet-based research. That is, research on understanding core VoIP issues and receiver-based mechanisms for optimising the perceivable quality. This seemed to be the case both during the frantic Integrated Services and Differentiated Service periods. The first RFC for RTP appeared as late as 1996 [121] even though RTP was being used in the VIC and VAT tools since 1992.

As for speech coding, the first G.729 standard was released in 1996 [62]. As noted earlier, G.729 is an 8 kb/sec LPC-based coder still used in many VoIP applications today. This includes the Skype application when using IP to telephony services e.g. in the SkypeIn and SkypeOut services. As the load on the Internet grew, studies of error recovery were being published [13, 110]

Two forwarding-looking articles were also published in 1997, [79] and [23]. The first suggested that the research community should concentrate on: quality issues for voice, in particular the effect of consecutive losses on speech quality, RTP multiplexing, and multicast. The second details how the Internet needs to be modified to host IP telephony applications.

The ITU-T E-model was first proposed in 1998 [64]. Some important IETF standards were also first published in the same year namely RTSP [122] and SDP [50]. Some methods for recovering lost VoIP packets are summarised in [104], with FEC techniques summarised in [105].

4.5 Internet telephony comes of age (2000-present)

As IP voice entered the mainstream, Internet telephony research became more focused as well as standardised. We will begin with the standardisation of Internet telephony, then move onto the research efforts.

During the past nine years, the signalling protocols have become established, SIP and H.323, continue to be developed. Today, SIP also plays a central role in IMS [19]. In the non-standard protocol realm some researchers have reverse engineered the Skype signalling protocol and published their findings [131, 45].

One of the more active research areas has been in wireless voice services. Focus has mainly been in the areas of throughput and capacity issues of IEEE 802.11 networks. Casetti et al. present a framework that assumes variable rate speech coders at rates of 64 kb/s, 13 kb/s, and 8 kb/s [20]. Their rates are determined by an end to end control mechanism, based on measurements of packet delay and loss rates. Another approach is to look at the MAC protocol directly. Dong et al. propose and examine selective error checking (SEC) at the MAC layer of 802.11 [29]. They make use of the fact that speech bits can tolerate errors, but should be protected for optimal quality reproduction. Simulation results showed that the speech quality can be substantially improved by modifying the MAC layer with SEC to suit the Narrow-Band Adaptive MultiRate (NB-AMR).

Filali looks at a MAC tuning approach [33]. He exploits the properties of multimedia applications in IEEE 802.11-based wireless networks by limiting the number of retransmissions of a data frame by a source until the reception of a link-level acknowledgement from the destination.

In 2005, the IEEE approved QoS service enhancements for local area network applications called IEEE 802.11e. Garg et al. examines using the IEEE 802.11e protocol for voice applictions [36]. The Enhanced Distributed Coordination Function (EDCF) has been proposed as a MAC protocol. EDCF assigns four different priority classes for incoming packets at each node which are called access categories (AC). Each AC has its own channel access function. This is in contrast to the standard Distributed Coordination Function (DCF) where packets all use the same access function to the channel. Access functions for different categories means assigning delay times, minimum contention windows, and the number of back-off stages for each type of service.

Garg et al. looked at 802.11e's ability to fulfil the goals of improved QoS and higher channel efficiency. They investigated the response of the protocol to options in the protocol parameters and showed that the Hybrid Coordination Function (HCF) reduces channel contention and provides improved channel utilisation. Both MAC coordination functions, EDCF and HCF, are sensitive to protocol parameters which are dependent on the scheduling algorithms. They conclude that further investigations need to be conducted.

Kawata et al. propose a dynamic Point Coordinator Function (PCF) for improved capacity [81]. They suggest two new media access schemes, dynamic point coordination function (DPCF) and modified DPCF (DPCF2). The claim is that the capacity of VoIP traffic can be increased by up to 20% in 802.11b networks. They show how a significant improvement in the end-to-end delay with mixed VoIP and data traffic can be achieved. Delay is maintained at approximately 100 ms in heavily loaded traffic conditions, whilst at 60 ms in normal traffic conditions.

Lindgren et al. [88] evaluate four mechanisms for providing service differentiation in IEEE 802.11 networks. The evaluated schemes are the PCF of IEEE 802.11, EDCF of IEEE 802.11e extension, Distributed Fair Scheduling (DFS), and Blackburst. Using simulation they looked at throughput, medium utilisation, collision rate, average access delay, and delay distribution for a variable load of real time and background traffic. The simulations showed that the best performance is achieved by Blackburst. PCF and EDCF are also able to provide good service differentiation. DFS can provide relative differentiation and consequently avoids starvation of low priority traffic.

Currently voice occupies relatively little of the IP wireless access capacity and the majority of voice traffic is carried by the cellular networks. Research in combining these two has been published within the context of voice roaming [17, 92]. Exploring voice quality in IP networks continues to be an active research area [41, 137].

Bibliography

[1] ITU-T Study Group 12 Delayed Contribution 27. General Prediction of the Impairment due to Dependent (Non-Random) Packet Loss for Inclusion in the E-model, January 2005. ITU-T SG 12.

[2] Chris Adams and Stephen Ades. Voice experiments in the UNIVERSE project. In *IEEE Record of the International Conference on Communications (ICC)*, pages 927–935, Chicago, Illinois, June 1985.

[3] Paul Almquist. Type of service in the Internet protocol suite. RFC 1349, Internet Engineering Task Force, July 1992.

[4] Soren Vang Andersen. iLBc - a linear predictive coder with robustness to packet losses. In *IEEE Workshop on speech coding*, pages 23–25, October 2002.

[5] Giulio Barberis. Buffer Sizing of a Packet-Voice Receiver. *IEEE Transactions on Communications*, 29(2):152–156, February 1981.

[6] Giulio Barberis and Daniele Pazzaglia. Analysis and Optimal Design of a Packet-Voice Receiver. *IEEE Transactions on Communications*, 28(2):217–227, February 1980.

[7] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen. The adaptive multirate wideband speech codec (AMR-WB). *IEEE Trans. on Speech and Audio Processing*, 10(8):620–636, November 2002.

[8] Uyless Black. *Internet Telephony: Call Processing Protocols*. Prentice-Hall, November 2000.

[9] Steven Blake et al. An architecture for differentiated services. *Request for Comments (Informational) RFC 2475, Internet Engineering Task Force*, December 1998.

[10] Jean Bolot. Characterizing end-to-end packet delay and loss in the Internet. *Journal of High Speed Networks*, 2(3):305–323, 1993.

[11] Jean Bolot. End-to-end packet delay and loss behavior in the Internet. In Deepinder Sidhu, editor, *ACM Symposium on Communications Architectures and Protocols*, pages 289–298, San Francisco, California, September 1993.

[12] Jean Bolot, Hugues Crepin, and Andrés Vega-Garcia. Analysis of audio packet loss in the internet. In *Proceedings International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Lecture Notes in Computer Science, pages 163–174, Durham, New Hampshire, April 1995. Springer.

[13] Jean Bolot and Andrés Vega-Garcia. The case for FEC-based error control for packet audio in the Internet. In *ACM Multimedia Systems*, 1997.

[14] Catherine Boutremans. *Delay Aspects in Internet Telephony.* PhD thesis, EPFL, December 2003.

[15] Robert Braden, David Clark, and Scott Shenker. Integrated services in the Internet architecture: An overview. *Request for Comments (Informational) RFC 1633, Internet Engineering Task Force*, June 1994.

[16] Kim Byoung-Jo, R. Shankaranarayanan, N.K. Henry, P.S. Schlosser, and K. Fong. The AT & T Labs broadband fixed wireless field experiment. *IEEE Communications Magazine*, 37(10):56–62, October 1999.

[17] Andrea Calvagna, Giacomo Morabito, and A. Pappalardo. WiFi mobility framework supporting GPRS roaming: Design and Implementation. In *IEEE International Conference on Communications*, pages 116–120, 2003.

[18] Gonzalo Camarillo. *SIP Demystified.* McGraw-Hill Professional, August 2001.

[19] Gonzalo Camarillo and Miguel-Angel Garca-Martn. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds.* Wiley, February 2006.

[20] Claudio Casetti and Carla-Fabiana Chiasserini. Improving fairness and throughput for voice traffic in 802.11e EDCA. In *IEEE PIRMC'04*, Barcelona, Spain, September 2004.

[21] Stephen L. Casner and S. E. Deering. First IETF Internet Audiocast. *ACM Computer Communication Review*, 22(3):92–97, July 1992.

[22] Alan Clark, Robert Cole, and Kaynam Hedayat. RTCP extensions for voice over IP metric reporting. *Internet Draft, draft-clark-avt-rtcpvoip-01.txt*, July 2002.

[23] David Clark. A Taxonomy of Internet Telephony Applications. In *25th Telecommunications Policy Research Conference*, Washington, DC, September 1997.

[24] Danny Cohen. Issues in Transnet Packetized Voice Communications. In *Proceedings of the Fifth Data Communications Symposium*, pages 6–10–6–13, Snowbird, Utah, September 1977.

[25] Dany Cohen. Realtime networking and packet voice. In *ACM Sigcomm Tutorial*, August 1999.

[26] SIP / H.323 Comparison. Numera. http://www.nuera.com/applications/sipH323pfv.cfm.

[27] ITU-T Delayed Contribution D.221. E-Model: Additivity of Burst Packet Loss Impairment with other Impairment Types, March 2004.

[28] Ismail Dalgic and Hanlin Fang. Comparison of H.323 and SIP for IP telephony signaling. In *Photonics East*, Boston, Massachusetts, September 1999. SPIE.

[29] Hui Dong, I.D. Chakares, A Gersho, E. Belding-Royer, and J.D. Gibson. Selective bit-error checking at the MAC layer for voice over mobile ad hoc networks with IEEE 802.11. In *WCNC*, March 2004.

[30] The Economist. The end of the line, October 2006.

[31] European Telecommunications Standards Institute. Generic Access Network (GAN). *3GPP TS 43.318*, 2005.

[32] Domenico Ferrari and Dinesh C. Verma. A Scheme For Real-Time Channel Establishment In Wide-Area Networks. *IEEE Journal On Selected Areas In Communications*, 8(3):368–379, 1990.

[33] Fethi Filali. Dynamic and efficient tuning of IEEE 802.11 for multimedia applications. In *IEEE PIMRC 04*, pages 910–914, Barcelona, Spain, September 2004.

[34] James W. Forgie. Speech communications in packet-switched networks. *91st Journal of the Acoustic Society of America*, 59(1), April 1976.

[35] Timur Friedman, Ramon Caceres, and Alan Clark. RTP extended reports (rtp xr). *IETF Internet Draft (work in progress), draft-ietf-avt-rtcp-report-extns-05.txt*, April 2003.

[36] P. Garg, R. Doshi, R. Greene, M. Baker, M. Malek, and X. Cheng. Using IEEE 802.11e MAC for QoS over Wireless. In *Proceedings of*

the 22nd IEEE International Performance Computing and Communications Conference (IPCCC 2003), Phoenix, Arizona, April 2003.

[37] Gerald Q. Maguire Jr. Practical Voice Over IP (VoIP): SIP and related protocols. In *http://www.it.kth.se/courses/IK2554/VoIP-Coursepage-Fall-2009.html*, 2009.

[38] Bernard Gold. Digital speech networks. *Proceedings of the IEEE*, 65(12):1636–1658, December 1977.

[39] Charles Goodwin. *Conversational organization: Interaction between speakers and hearers.* Academic Press, New York, 1981.

[40] Prabandham M. Gopal and Barath Kadaba. A simulation study of network delay for packetized voice. In *Proc. IEEE Global Telecommunications Conf. (GLOBECOM)*, December 1986.

[41] Volodya Grancharov. *Human Perception in Speech Processing.* PhD thesis, Royal Institute of Technology (KTH), June 2006. TRITA-EE 2006:016.

[42] Robert M. Gray. The 1974 origins of VoIP. *IEEE Signal Processing Magazine*, 22(4):87–90, July 2005.

[43] John G. Gruber. Delay Related Issues in Integrated Voice and Data Networks - A Review and Some Experimental Work. In *6th Data Communications Symposium*, pages 166–180, Pacific Grove, California, November 1979.

[44] John G. Gruber and Leo Strawczynski. Subjective Effects Of Variable Delay and Speech Clipping In Dynamically Managed Voice Systems. *IEEE Transactions on Communications*, COM-33(8):801–808, 1985.

[45] Saikat Guha, Neil Daswani, and Ravi Jain. An Experimental Study of the Skype Peer-to-Peer VoIP System. In *IPTPS*, 2006.

[46] Marie Guguin, Valrie Gautier-Turbin, Latitia Gros, Vincent Barriac, Rgine Le Bouquin-Jeann, and Grard Faucon. Study of the relationship between subjective conversational quality, and talking, listening and interaction qualities: Towards an objective model of the conversational quality. In *Proceedings of the Measurement of Speech and Audio Quality in Networks workshop (MESAQIN'05)*, Prague, Czech Republic, June 2005.

[47] Olof Hagsand, Ian Marsh, and Kjell Hanson. Sicsophone: A Low-Delay Internet Telephony Tool. In *IEEE 29th Euromicro Conference*, pages 189–195, Belek, Turkey, September 2003.

[48] Florian Hammer, Peter Reichl, and Alexander Raake. Elements of Interactivity in Telephone Conversations. In *8th International Conference on Spoken Language Processing (ICSLP/INTERSPEECH 2004)*, pages 1741–1744, Jeju Island, Korea, October 2004.

[49] Florian Hammer, Peter Reichl, and Alexander Raake. The well-tempered conversation: Interactivity, delay and perceptual VoIP quality. In *IEEE ICC 2005*, Seoul, Korea, 2005.

[50] Mark Handley and Van Jacobson. SDP: Session Description Protocol. *Request for Comments (Standards Track) RFC 2327, Internet Engineering Task Force*, April 1998.

[51] Mark Handley, Henning Schulzrinne, Eve Schooler, and Jonathan Rosenberg. SIP: Session initiation protocol. *Request for Comments (Standards Track) RFC 1883, Internet Engineering Task Force*, March 1999.

[52] Vicky Hardman, Angela Sasse, Mark Handley, and Anna Watson. Reliable Audio for Use over the Internet. In *Proceedings of INET'95*, Honolulu, Hawaii, June 1995.

[53] Olivier Hersent, David Gurle, and Jean-Pierre Petit. *IP telephony.* Addison Wesley, Reading, Massachusetts, 2000.

[54] Christian Hoene. *Internet Telephony over Wireless Links.* PhD thesis, Technical University of Berlin, Germany, September 2005.

[55] Christian Hoene and Georg Carle. Umts networks and internet telephony. http://net.informatik.uni-tuebingen.de/en/teaching/umts-voip/ss2007.

[56] Christian Hoene, B. Rathke, and Adam Wolisz. On the Importance of a VoIP Packet. In *Proceedings Of 1st ISCA Tutorial and Research Workshop On The Auditory Quality Of Systems*, Mont-Cenis, Germany, April 2003.

[57] J. M. Holtzman. The interaction between queueing and voice quality in variable bit rate packet voice systems. In Minoru Akiyama, editor, *Eleventh International Teletraffic Congress*, pages 151–154, Kyoto, Japan, September 1985. Elsevier Science Publishers.

[58] International Telecommunication Union. 7 kHz audio-coding within 64 kbit/s. *ITU-T Recommendation G.722*, November 1988.

[59] International Telecommunication Union. Echo supressors. *ITU-T Recommendation G.164*, November 1988.

[60] International Telecommunication Union. Specification for an intermediate reference system. *ITU-T Recommendation P.48*, November 1988.

[61] International Telecommunication Union. Annex B: A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70. *ITU-T Recommendation G.729 Annex B*, November 1996.

[62] International Telecommunication Union. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP). *ITU-T Recommendation G.729*, March 1996.

[63] International Telecommunication Union. Modulated noise reference unit (MNRU). Technical Report ITU-T Recommendation P.810, Telecommunication Standardization Sector of ITU, 1996.

[64] International Telecommunication Union. The E-model, a computational model for use in transmission planning. Recommendation G.107, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, December 1998.

[65] International Telecommunication Union. Pulse Code Modulation (PCM) of Voice Frequencies. *ITU-T Recommendation G.711*, November 1998.

[66] International Telecommunication Union. Appendix I: A high quality low-complexity algorithm for packet loss concealment with G.711. *ITU-T Recommendation G.711, Appendix I*, September 1999.

[67] International Telecommunication Union. Single ended method for objective speech quality assessment in narrow-band telephony applications. *ITU-T Recommendation P.563*, 2004.

[68] IPTEL. SIP versus H.323. http://www.iptel.org/info/trends/sip.html.

[69] ITU-T Recommendation H.323. Packet-based multimedia communications systems, July 2003.

[70] Van Jacobson. Multimedia conferencing on the Internet. In *ACM Symposium on Communications Architectures and Protocols*, London, England, August 1994. Tutorial slides.

[71] Van Jacobson and Steve McCanne. vat - LBNL Audio Conferencing Tool, July 1992. Available at http://www-nrg.ee.lbl.gov/vat/.

[72] Nuggehally S. Jayant and Susan W. Christensen. Effects of packet losses in waveform coded speech and improvements due to an odd-even

sample-interpolation procedure. *IEEE Transactions on Communications*, 29(2):101–109, February 1981.

[73] Wenyu Jiang and Henning Schulzrinne. Analysis of On-Off Patterns in VoIP and Their Effect on Voice Traffic Aggregation. In *9th IEEE International Conference on Computer Communication Networks*, Las Vegas, Nevada, October 2000.

[74] Wenyu Jiang and Henning Schulzrinne. Modeling of Packet Loss and Delay and their Effect on Real-Time Multimedia Service Quality. In *Proceedings International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, June 2000.

[75] Wenyu Jiang and Henning Schulzrinne. Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss. In *Proceedings International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Miami Beach, Florida, May 2002.

[76] Wenyu Jiang and Henning Schulzrinne. Speech Recognition Performance as an Effective Perceived Quality Predictor. In *IWQoS*, Miami Beach, May 2002.

[77] Jonathan Rosenberg and Henning Schulzrinne. SIP: Comparison of SIP and H.323. In *Proceedings of NOSSDAV*, Cambridge, UK, July 1998.

[78] Jonathan Rosenberg and R. Mahy and P. Matthews and D. Wing. Session Traversal Utilities for (NAT) (STUN), July 2008.

[79] Jonathan Rosenburg. Internet Telephony: A (Partial) Research Agenda, October 1997.

[80] Lars-Erik Jonsson. RObust Header Compression (ROHC): The ROHC Architecture. *IETF Internet Draft, draft-jonsson-rohc-architecture-00.txt*, December 2002.

[81] Takehiro Kawata, S. Shin, Andrea G. Forte, and Henning Schulzrinne. Using dynamic PCF to improve the capacity for VoIP traffic in IEEE 802.11 networks. In *IEEE WCNC*, March 2005.

[82] Moo Young Kim and W. Bastiaan Kleijn. Rate-Distortion comparisons between FEC and MDC based on Gilbert channel model. In *Proc. IEEE Int. Conf. on Networks (ICON)*, pages 495–500, Sydney, 2003.

[83] Nobuhiko Kitawaki, M. Honda, and K. Itoh. Speech-quality assessment methods for speech-coding systems. *IEEE Communications Magazine*, pages 26–32, October 1984.

[84] Bastiaan Kleijn and Kuldip. K. Paliwal. *Speech Coding and Synthesis*. Amsterdam: Elsevier, 1995.

[85] Vineet Kumar, Markku Korpi, and Senthil Sengodan. *IP Telephony with H.323: Architectures for Unified Networks and Integrated Services*. Wiley, March 2001.

[86] H. H. Lee and C. K. Un. A study of on-off characteristics of conversational speech. *IEEE Transactions on Communication*, 34(6):630–637, June 1986.

[87] Fengyi Li. Measurements of Voice over IP Quality. Master's thesis, KTH, Royal Institute of Technology, Sweden, 2002.

[88] Anders Lindgren, Andreas Almquist, and Olov Schelén. Quality of Service Schemes for IEEE 802.11 Wireless LANs - An Evaluation. *In the Journal on Special Topics in Mobile Networking and Applications (MONET) on Performance Evaluation of Qos Architectures in Mobile Networks*, 8:223–235, June 2003.

[89] M. R. Schroeder and B. S. Atal. Code-excited linear prediction (CELP): high-quality speech at very low bit rates. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '85)*, pages 937–940, April 1985.

[90] Andrew J. Mackie, Salah E. Aidarous, Samy A. Mahmoud, and J. Spruce Riordon. Design and performance evaluation of a packet voice system. *IEEE Transactions on Vehicular Technology*, VT-32(2):158–168, May 1983.

[91] D. T. Magill. Adaptive speech compression for packet communication systems. In *IEEE National Telecommunications Conference*, pages 29D–1–29D–5, 1973.

[92] Ian Marsh, Björn Grönvall, and Florian Hammer. The design and implementation of a quality-based handover trigger. In *Proceedings Of The 5th IFIP-TC6 Networking Conference*, Coimbra, Portgual, May 2005.

[93] Piet Van Mieghem. A lower bound for the end-to-end delay in networks: Application to voice over IP. In *IEEE Globecom*, pages 2508–2513, Sydney, Australia, November 1998.

[94] Warren A. Montgomery. Techniques for Packet Voice Synchronization. *IEEE Journal on Selected Areas in Communications*, SAC-1(6):1022–1028, December 1983.

[95] Sue B. Moon, Jim Kurose, and Don Towsley. Packet Audio Playout Delay Adjustment Algorithms: Performance Bounds and Algorithms. Research report, Department of Computer Science, University of Massachusetts, Amherst, Massachusetts, August 1995.

[96] Nanying Yin and Thomas E. Stern and Song Li. Performance Analysis of a Priority-Oriented Packet Voice System. In *IEEE Infocom*, pages 856–863, San Francisco, California, April 1987.

[97] William Edward Naylor. A status report on the real-time speech transmission work at UCLA. NSC Note 52, December 1974.

[98] William Edward Naylor. *Stream traffic communication in packet switched networks*. PhD thesis, UCLA, August 1977.

[99] William Edward Naylor and Leneord Kleinrock. Stream traffic communication in packet switched networks: destination buffering considerations. *IEEE Transactions on Communications*, 30:2527–2534, 1982.

[100] Anders Gunnar (n'ee Andersson). Capacity Study of Statistical Multiplexing for IP Telephony. Master's thesis, Uppsala University, 2000.

[101] Olivier Hersent and Jean-Pierre Petit and David Gurle. *Deploying Voice-over-IP Protocols*. Wiley, 2005.

[102] Jorg Ott and E. Carrara. Extended RTP Profile for RTCP-based Feedback (RTP/AVPF). *IETF Request for comments 4585*, July 2006.

[103] Packetizer. H.323 versus SIP: A Comparison. http://www.packetizer.com/iptel/h323_vs_sip.

[104] Charlie E. Perkins, Orion Hodson, and Vicky J. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Audio. *IEEE Network*, 12(5):40–48, September 1998.

[105] Mathew Podolsky, Cynthia Romer, and Steve McCanne. Simulation of FEC-Based Error Control for Packet Audio on the Internet. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, volume 2, pages 505–512, San Francisco, California, March 1998.

[106] Alexander Raake. *Speech Quality of VoIP: Assessment and Prediction*. John Wiley & Sons, 2006.

[107] Peter Reichl and Florian Hammer. Hot discussion or frosty dialogue? Towards a temperature metric for conversational interactivity. In *8th International Conference on Spoken Language Processing (ICSLP/INTERSPEECH 2004)*, Jeju Island, Korea, October 2004.

[108] Antony W. Rix. Comparison between subjective listening quality and P.862 PESQ score. In *Proc. Measurement of Speech and Audio Quality in Networks (MESAQIN'03)*, Prague, Czech Republic, May 2003.

[109] Rodeo group. http://www-sop.inria.fr/rodeo/fphone/, 1999.

[110] J. Rosenberg and H. Schulzrinne. An RTP payload format for generic forward error correction. *Request for Comments (Standards Track) RFC 2733, Internet Engineering Task Force*, December 1999.

[111] Jonathan Rosenberg, R. Mahy, and Christian Huitema. Traversal Using Relay NAT (TURN).

[112] Jonathan Rosenberg, Lili Qiu, and Henning Schulzrinne. Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithms on the Internet. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 1705–1714, Tel Aviv, Israel, March 2000.

[113] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, A. Johnston, J. Peterson, R. Sparks, Mark Handley, and Eve Schooler. RFC 3261 - SIP: Session initiation protocol. Technical report, Internet Engineering Task Force, June 2002.

[114] Jonathan Rosenberg, J. Weinberger, Christian Huitema, and R. Mahy. STUN – Simple Traversal of User Datagram Protocol (UDP) through Network Address Translators (NATs). Internet Engineering Task Force: RFC 3489, March 2003.

[115] Benny Sällberg, Henrik Åkesson, Nils Westerlund, Mattias Dahl, and Ingvar Claesson. Analog circuit implementation for speech enhancement purposes. In *Proc. 38th Asilomar Conference on Signals, Systems, and Computers*, volume 2, pages 2285–9, CA, USA, 2004.

[116] Benny Sällberg and Mattias Dahl. Speech enhancement implementations in the digital, analog, and hybrid domain. In *Swedish System-on-chip Conference*, Tammsvik, Stockholm, 2005.

[117] Salman A. Baset and Henning Schulzrinne. An Analysis of the Skype Peer-to-Peer Internel Telephony Protocol. In *Proceedings of the IEEE Infocom Conference*, Barcelona, Spain, April 2006.

[118] Henning Sanneck. Fehlerverschleierungsverfahren für Sprachübertragung mit Paketverlust. Master's thesis, Telecommunications Department, University of Erlangen-Nuremberg, Germany, June 1995.

[119] Henning Schulzrinne. Voice Communication Across the Internet: A Network Voice Terminal. Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.

[120] Henning Schulzrinne. *Reducing and characterizing packet loss for high-speed computer networks with real-time services*. PhD thesis, University of Massachusetts, Amherst, Massachusetts, May 1993.

[121] Henning Schulzrinne et al. RTP: A transport protocol for real-time applications. *Request for Comments (Standards Track) RFC 3550, Internet Engineering Task Force*, July 2003.

[122] Henning Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (RTSP). *Request for Comments (Standards Track) RFC 2326, Internet Engineering Task Force*, April 1998.

[123] Scott Shenker. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications*, 13(7):1176–1188, September 1995.

[124] Christian Sieckmeyer. Bewertung von adaptiven Ausspielalgorithmen fr paketvermittelte Audiodaten Evaluation of adaptive playout algorithms for packet audio - in German. Studienarbeit, Dept. of Electrical Engineering, TU Berlin, Berlin, Germany, October 1995.

[125] Henry Sinnreich and Alan B. Johnston. *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol (Networking Council)*. Wiley, July 2006.

[126] Skype Communications S.A. Skype Explained, 2007.

[127] Kotikalapudi Sriram and Ward Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):833–846, September 1986.

[128] Thomas E. Stern. A queueing analysis of packet voice. In *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, pages 2.5.1–6, San Diego, California, November/December 1983.

[129] Donald L Stone and S. Jeffay. An Empirical Study Of Delay Jitter Management Policies, 1995.

[130] Tatsuya Suda, Yechiam Yemini, Hideo Miyahara, and Toshiharu Hasegawa. Performance Evaluation of a Packetized Voice System - A Simulation Study. In *IEEE ICC*, pages 749–753, Boston, Massachusetts, June 1983.

[131] Kyoungwon Suh, Daniel R. Figueiredo, Jim Kurose, and Don Towsley. Characterizing and Detecting Skype-Relayed Traffic. In *IEEE INFOCOM 2006 - The Conference on Computer Communications*, 2006.

[132] Krister Svanbro. Lower layer guidelines for robust RTP/UDP/IP header compression. *IETF Internet Draft (work in progress), draft-ietf-rohc-rtp-lower-layer-guidelines-03.txt*, December 2001.

[133] Thomas E. Tremain. The government standard linear predictive coding algorithm: LPC-10. *Speech Technology*, 1:40–49, April 1982.

[134] International Telecommunication Union. Transmission Systems and Media, General Recommendation on the Transmission Quality for an Entire International Telephone Connection; One-Way Transmission Time. *G.114*, March 1993.

[135] International Telecommunication Union. Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *ITU-T Recommendation P.862*, February 2001.

[136] Mártin Varela. *Pseudo-Subjective Quality Assessment of Multimedia Streams and its Applications in Control*. PhD thesis, University of Rennes, November 2005.

[137] Martín Varela. Studying the Effects of FEC on Voice Traffic Using PSQA (extended abstract). In *IEEE INFOCOM 2005 Student Workshop*, Miami, USA, March 2005.

[138] Dinesh C. Verma, Hui Zhang, and Domenico Ferrari. Delay jitter control for real-time communication in a packet switching network. Technical Report TR-91-007, University of California, Berkeley, CA, 1991.

[139] M. A. Visser and Magda El Zarki. Voice and data transmission over an 802.11 wireless network. In *Proceedings of IEEE PIMRC'95*, pages 648–652, Toronto, Canada, September 1995.

[140] Clifford J. Weinstein and James W. Forgie. Experience with Speech Communication in Packet Networks. *IEEE Journal on Selected Areas in Communications*, SAC-1(6):963–980, December 1983.

Appendix: Included articles

Paper A

Bengt Ahlgren, Anders Andersson, Olof Hagsand, and Ian Marsh.
Dimensioning Links for IP Telephony. In *Proceedings of the 2nd IP-Telephony Workshop*, pages 14-24, New York, USA, April 2001.

"No-one ever said it was gonna be easy"

Inspiral Carpets (feat. Mark E. Smith) - I want you

Dimensioning Links for IP Telephony

Bengt Ahlgren[1], Anders Andersson[1], Olof Hagsand[2] and Ian Marsh[1]

[1] SICS, CNA Laboratory, Sweden
bengta@sics.se, anders@sics.se, ianm@sics.se
[2] LCN Laboratory, IMIT, Royal Institute of Technology, Sweden
olof@sics.se

Abstract. Packet loss is an important parameter for dimensioning network links or traffic classes carrying IP telephony traffic. We present a model based on the Markov modulated Poisson process (MMPP) which calculates packet loss probabilities for a set of superpositioned voice input sources and link properties. We do not introduce another new model to the community, rather try and verify one of the existing models via extensive simulation and a real world implementation. A plethora of excellent research on queuing theory is *still* in the domain of ATM researchers, hence we attempt to highlight their validity to the IP (Telephony) community.
Packet level simulations show reasonable correspondence with the predictions of the model. Our main contribution is the verification of the MMPP model with measurements in a laboratory environment. The loss rates predicted by the model are in general close to the measured loss rates and the loss rates obtained by simulation. The general conclusion is that the MMPP-based model is a tool well suited for dimensioning links carrying packetised voice in a system with limited buffer space.

1 Introduction

Voice applications, such as telephony, have been used on the best effort service provided by the Internet for some time. Currently many telephone operators have plans to use IP technology as a bearer also for the regular telephone service. This, however, requires that the IP network can provide service guarantees. Quality of Service (QoS) issues are being addressed by many forums, committees and researchers. Research on IP QoS has concentrated on the issues of classifying, scheduling and admission of packets into a network. Less has been done on how to dimension an IP network carrying real time traffic. This paper focuses on dimensioning IP networks intended to carry or voice calls. It is feasible that existing carriers would like to allocate a portion of their bandwidth for this service and through mechanisms such as differentiated services [NJZ97], provide superior service for this kind of data and optionally levy higher charges.

Our approach is to look at work done in *both* the ATM and traditional telephony communities as well as to use tools and simulators from the IP community to verify the ideas in an environment relevant for the Internet today. We have seen very little work which has taken this approach.

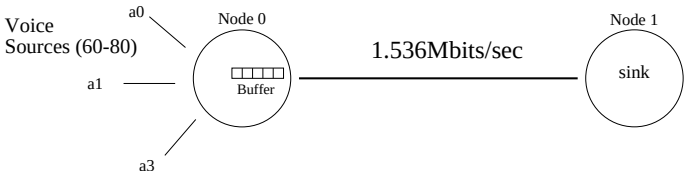

Fig. 1. Dimensioning a link for voice sources over IP networks

Figure 1 illustrates the problem scenario we are addressing. A number of packet voice sources are multiplexed onto a link. The link has a limited amount of buffering which sometimes will result in the loss of packets with consequences on the sound quality. With a link of a given bandwidth and a number of voice sources, what kind of quality could be expected if we ran 60 sources? What if we increased the number of sources to 80, can we still expect adequate quality? How will we affect the system by changing the amount of buffering in the router?

We present a mathematical model based on a Markov modulated Poisson process (MMPP) which can predict the packet loss probability. We first verify the model using the `ns-2` packet level simulator. The main contribution of this paper is the verification of the MMPP model with measurements in a lab network. These experiments show a reasonable correspondence between the loss rate predicted by the model and the loss rate measured in the lab.

The rest of the paper is organized as follows. After summarizing relevant related work in the next section, we present the MMPP-based mathematical model. Section 3. Section 4 describes the parameters we used in the experiments. Sections 5 and 6 describe the `ns-2` simulations and the laboratory experiments, respectively. The experimental results are presented and discussed in Section 7, the paper is concluded with Section 8.

2 Related work

Link dimensioning for voice has been a research topic for several decades in both academia and the telecommunications industry. Starting a little more than ten years back, the research focus has been on link dimensioning for ATM networks. Most of the results in the domain of ATM networks are also applicable in the domain of IP networks. The majority of the results from previous research is theoretical or results from simulations. Our research also adds results of measurements from a real system.

Several approaches have been suggested in the literature to solve the problem of dimensioning links in packet switched networks. Anick, Mitra and Sondhi [AMS82] study a multiplexer with infinite buffer using a *stochastic fluid flow* model but it is shown by Zheng [Sun98] that this model only works for a multiplexer under heavy load. Tucker [Tuc88] studies a multiplexer with finite buffer using the fluid flow model, but it does not fit the model for small buffers. Heffes and

Lucantoni [HL86] uses a two-state *Markov modulated Poisson process* (MMPP) successfully to estimate the delay in a multiplexer with infinite buffer size. They suggest that the same approach for calculating the parameters of the MMPP can be used for a multiplexer with finite buffer size, but Nagarajan, Kurose and Towsley [NKT91] show that this does not work in the case of finite buffer size. Instead, they develop a different method for finding the parameters of the MMPP. Baiocchi *et al.* [BML+91] approximate the arrival process with a two-state MMPP and suggest a method called *asymptotic matching* for the calculation of the parameters of the MMPP. This approach is used by Andersson [And00] together with a procedure to calculate the loss probabilities developed by Baiocchi, Melazzi and Roveri [BBMR91] to study a multiplexer loaded with a superposition of voice sources.

3 A mathematical model

In this section we develop a mathematical model for dimensioning a link carrying voice traffic. We start with the arrival process of a single IP telephony source and proceed with the superposition of independent identically distributed sources. The sources are then multiplexed on a bottleneck link through a queue of limited size. A more detailed description of this model can be found in [And00]. The model is based on a model developed by Baiocchi, Melazzi and Roveri [BBMR91].

3.1 Single source properties

Most standard voice encodings have a fixed bit rate and a fixed packetization delay. They are thus producing a stream of fixed size packets. This packet stream is however only produced during talk-spurts—the voice coder sends no packets during silence periods.

The behavior of a single source is simply modeled by a simple on-off model (Figure 2). During talk-spurts (ON periods), the model produces a stream of

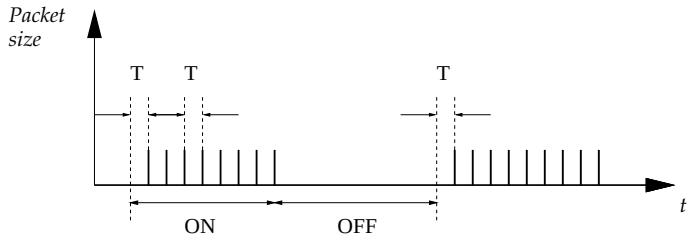

Fig. 2. Characteristics of a single source.

fixed size packets with fixed inter-arrival times T. Note that the first packet is produced one packet time after the start of an ON period. This is the result

of the packetization—the voice coder has to collect voice samples before it can produce the first packet.

The number of packets in a talk-spurt, denoted with the stochastic variable N_b, is assumed to be geometrically distributed on the positive integers with mean n. This means that we can never have zero packets in a talk-spurt. This variant of the geometric distribution is sometimes called the *first success* distribution (see for instance Gut [Gut95, page 258]), and has the probability function:

$$P(N_b = k) = qp^{k-1},\ k = 1,\ 2,\ 3,\ \ldots \tag{1}$$

where q represents the probability that a packet is the last one in a talk-spurt. This means that $p = \frac{n-1}{n}$. This fact implies that the ON periods have an expected value of $\alpha = nT$, where n is the expected number of packets in a talk-spurt.

We assume that the OFF periods are exponentially distributed with mean β, which is documented and discussed by Sriram and Whitt [SW86]. A voice source may be viewed as a two state birth-death process with birth rate β and death rate α. The OFF state represents the idle periods whilst the ON state represents the talk-spurts. While in a talk-spurt, packets are generated with a rate of $\frac{1}{T}$ packets per second.

3.2 Approximating a single source

We have chosen to approximate the above model using exponentially distributed inter-arrival times with mean T instead of fixed inter-arrival times. The purpose of the approximation is to simplify the modelling of multiple sources.

We let $\tau \in Exp(\frac{1}{T})$ denote the stochastic variable which describes the inter-arrivals during talk-spurts, and N_b be the geometrically distributed stochastic variable with the probability function in Equation 1 its mean n being the number of packets in a talk-spurt. Moreover τ and N_b are assumed to be independent. It can be easily seen that the ON periods (denoted U) are exponentially distributed and that the mean length of a talk-spurt is the same as in the deterministic inter-arrival case (nT). Figure 3 illustrates the behaviour of a single source with

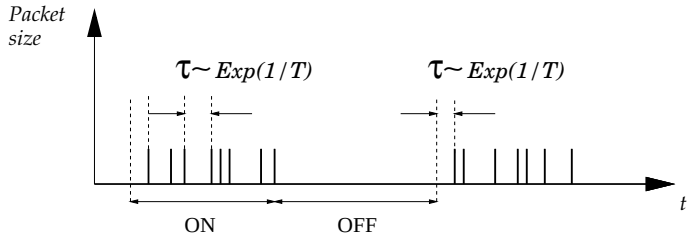

Fig. 3. A single source approximated with exponentially distributed inter-arrivals.

exponentially distributed inter-arrivals.

The OFF periods are assumed to be exponentially distributed with mean β. Because of the exponentially distributed inter-arrival times during a talk-spurt, the emission of packets during an ON period can be regarded as a Poisson process with intensity T. We can use the two state birth-death process to describe the packet generation with one state representing the idle periods and the other state representing the talk-spurts where packets are generated as a Poisson process with intensity T.

3.3 The superposition of independent voice sources

The superposition of voice sources can be viewed as a birth-death process where the states represent the number of sources that are currently in the ON-state. State i represents that i sources are active in a talk-spurt. We refer to the birth-death process as the *phase process* $J(t)$. The birth rate is given by the mean of the exponentially distributed idle periods, and we denote the mean as $\frac{1}{\beta}$. The death rate is determined by the mean of duration of the talk-spurts and is denoted $\frac{1}{\alpha}$. The probability p_{on} that a source is on is given by:

$$p_{\text{on}} = \frac{\alpha}{\alpha + \beta}.$$

3.4 The Markov modulated Poisson process

The *Markov modulated Poisson process* (MMPP) is a widely used tool for analysis of tele-traffic models (see, e.g., Heffes and Lucantoni [HL86]). It describes the superposition of sources of the type described in Section 3.2. When the phase process is in state i, i sources are on. The model graph of the MMPP is shown in Figure 4.

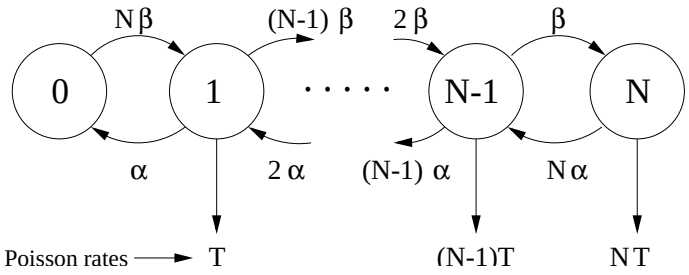

Fig. 4. Superposition of N voice sources with exponentially distributed inter-arrivals.

The superposition of Poisson processes is also a Poisson process. We can therefore simply add the intensities of the sources that are currently in a talk-spurt and obtain a new Poisson process for the superposition.

To validate the accuracy of approximating with a MMPP process, we calculated the index of dispersion of intervals (IDI) of multiple superpositioned

sources using a formula from Sriram and Whitt [SW86]. The IDI, also called the squared coefficient of variation, gives us some measure of how similar the traffic is in terms of burstiness. A y-value of 1 shows the traffic is as bursty as Poisson traffic, whereas a y-value of 18 is the burstiness of a single voice source. The high value accounts for the fact that the source is indeed bursty. The time period under which one observes this behaviour needs to stabilise.

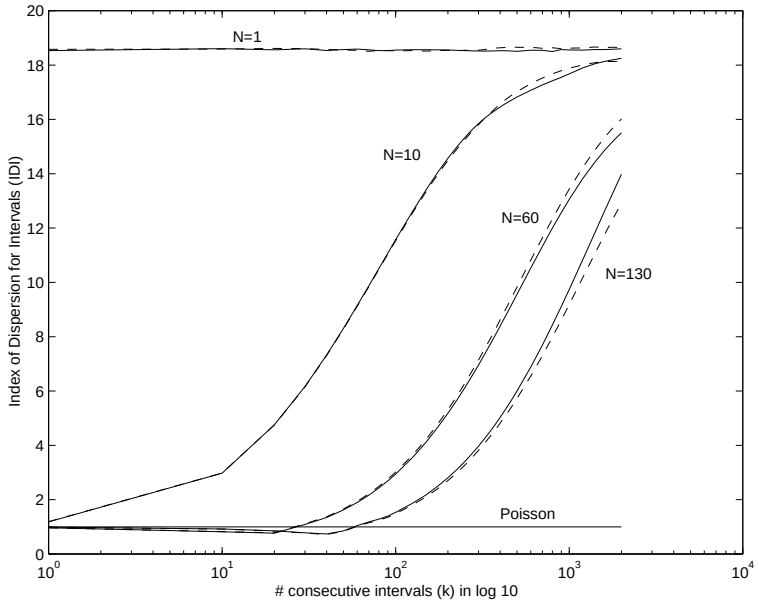

Fig. 5. k-interval squared coefficient of variation curves for the superposition of N voice sources.

Figure 5 shows c^2_{kN}, the IDI, against k for k between 1 and 2000 with between 1 and 130 sources, N, equal to 1, 10, 60 and 130. As a reference we have added the value of c^2_{kN} for a Poisson process. Data was obtained from simulations using Matlab. The solid line shows the c^2_{kN} for sources with deterministic inter-arrival times between packets during a talk-spurt, and the dashed lines show the c^2_{kN} for sources with exponentially distributed inter-arrival times, i.e., the MMPP approximation.

We see from the figure that the two descriptions of a single source behave in a similar way when they are superpositioned. The figure also shows that the superpositioned arrival process behaves as a Poisson process if we look at it for a short instant of time but it becomes much burstier if we study it over a longer period.

3.5 The multiplexer: MMPP/D/1/K queue

The arrival process described by the MMPP model is fed into a simple D/1/K queue. It is deterministic, has a single FIFO server and a buffer size (waiting room) which we vary. This kind of model is described in detail by Baiocchi *et al.* [BBMR91,BML+91]. We use their method and formulae for calculating the loss probabilities.

4 Parameter values

We used the following parameters in the MMPP model, simulations and lab experiments:

- 32 kb/s ADPCM voice encoding with 16 ms packet inter-arrival time, which results in 64 bytes of voice payload per packet
- A protocol header overhead consisting of 12 bytes for RTP, 8 bytes UDP and 20 bytes IP. We do not include any link layer headers. The resulting total packet size is 104 bytes, and the resulting bit rate is 52 kb/s.
- The number of successive packets in one talk-spurt is geometrically distributed on the positive integers with a mean of 22, which results in a mean talk-spurt length of 352 ms. The idle time between two successive bursts is exponentially distributed with a mean of 650 ms. The resulting average fraction of time a source is in a talk-spurt is 0.351.
- The bottleneck is a T1 link with a bandwidth of 1.536 Mb/s.

These values coincide with Sriram and Whitt [SW86] as well as previous work done by Zheng [Sun98] and Andersson [And00], except that we in this paper include protocol header overhead for the RTP/UDP/IP protocol stack.

Figure 6 shows loss curves computed with the MMPP model for a sample set of buffer sizes. The next step is to compare these loss probabilities from the model with results from `ns-2` simulations and measurements from a lab setup.

4.1 Load

We use between 60 and 80 sources to load the link. To define a load that is independent of the link bandwidth the load factor, or λ, is used:

$$Load\ (\lambda) = \frac{N \times P_{\text{on}} \times Rate_{\text{peak}}}{C}$$

where N is number of sources, C is the link capacity, P_{on} is the probability that the source is on and $Rate_{\text{peak}}$ speaks for itself. Table 1 shows loads for different numbers of sources.

Since 84 sources represents the number of sources where the mean bandwidth of the input equals the bandwidth of the link, we chose to use between 60 and 80 sources. The peak allocation yields just 29 sources (100 % utilisation when $P_{\text{on}} = 1$) so taking advantage of the probability that a source is off can yield more efficient link utilisation.

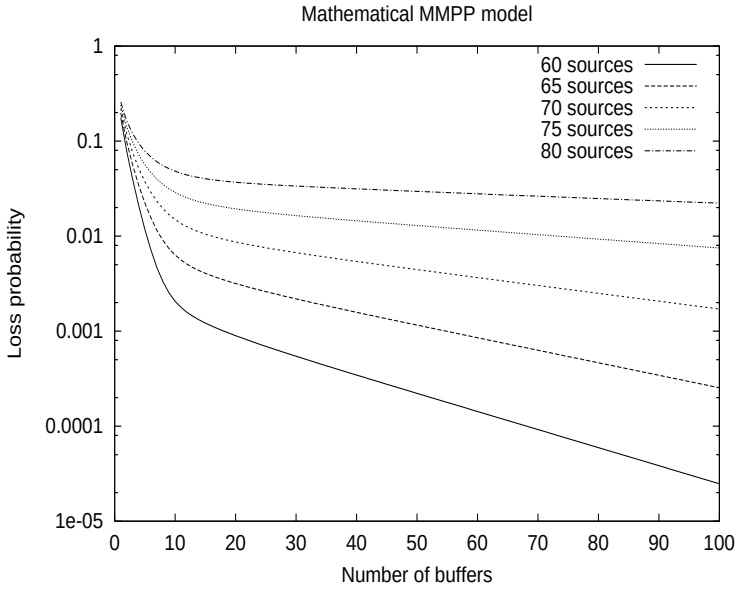

Fig. 6. Loss probabilities computed with the MMPP model.

Sources (N)	Load (λ)
29	34.5 %
60	71.4 %
80	95.3 %
84	98 %

Table 1. Network load for a number of sources.

4.2 Buffer size

We have chosen to simulate a multiplexer with an output link capacity of 1.536 Mb/s and buffer sizes ranging from 2 to 100 packets. With this choice of parameters we introduce a maximum queueing delay of 54 ms into the buffer. According to ITU recommendation G.114 [Int93] a delay of 0-150 ms acceptable for telephony, between 150 and 400 ms can also be acceptable, but over 400 ms is not. Therefore the total acceptable delay must be divided into a delay budget for each node in the path between the sender and receiver. If the path has 15 hops, and half of the delay budget can be allocated to queueing delay, we obtain 13.3 ms per hop. This translates into approximately 24 buffers per hop. For higher bandwidth links, the queueing delay per buffered packet decreases inversely proportional to the bandwidth.

5 ns-2 simulation

We used `ns-2` [FV98], a packet level simulator to verify the MMPP model. Figure 1 shows the topology used in the simulations and Figure 7 the Tcl snippet

```
set cbr($i) [new Agent/CBR/UDP]

set exp($i) [new Traffic/Expoo]
$exp($i) set packet-size 104
$exp($i) set burst-time 0.352s
$exp($i) set idle-time 0.65s
$exp($i) set rate 52K

$cbr($i) attach-traffic $exp($i)
```

Fig. 7. Tcl code fragment defining a source `ns-2`.

that is used to initialise "agents". They are constant rate sources, denoted by "CBR/UDP". Traffic/Expoo generates traffic based on an exponential on/off distribution with the parameters specified in the lower four lines. Each CBR source `$i$` uses a different random number seed, hence the sources will start independently of each other.

The simulation should run long enough for the system to reach steady state. A reasonable tradeoff is to use a simulated time of 1000 seconds in both the simulation and the lab experiments. 1000 seconds with an interval of 16 ms generates 22000 packets per source and 1.32 million packets for 60 sources or 1.76 million for 80 sources.

6 Lab network measurements

6.1 Topology

Figure 8 shows the experimental setup. A single machine acts as a traffic gen-

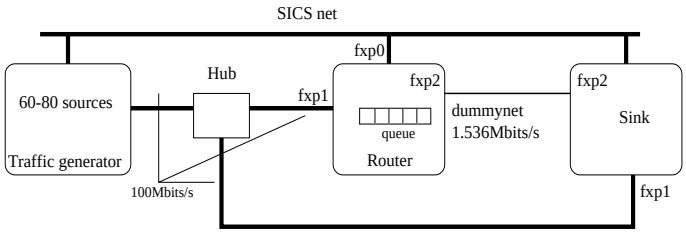

Fig. 8. Topology for Laboratory. The outgoing interface of the router is also connected to the sink.

erator and emulates several IP Telephony calls multiplexed together. The traffic is then sent on a shared 100 Mb/s Ethernet and received by two hosts: (1) a machine configured as a router; (2) a sink machine for measurements. The outgoing link of the router is connected to the sink. In this configuration the traffic is emitted by the generator, passes through the router and is received by the sink. Since the sink can observe the packets before it enters the router, it can directly find the latency and loss of each individual packet. The outgoing link of the router is constrained to 1.536 Mb/s using Dummynet [Riz97] which is explained in the next section. All the machines in the experiment were running FreeBSD 3.4.

6.2 Dummynet

Dummynet is a link emulator which allows arbitrary bandwidths and latencies to be specified to a virtual link. It is often used for emulating a slower link than what is physically available. Buffer sizes can be set for a given link and loss rates set to emulate the effect of lossy links. It is possible to create the illusion for TCP/UDP and IP that the link is like a WAN rather than a LAN. In this work, we are primarily interested in the lower bandwidth and configurable queue sizes. We modified the output functionality slightly to enable simpler calculation of the total number of packets received as well as the drop rate. Recording the total number of packets received gives us an additional check if the traffic generator or any system component dropped packets during the experiment.

The total number of sent packets remained the same for a given number of sources can be validated with the output of the traffic generator. It is trivial to divide the loss by the total number of packets in order to obtain the loss rate.

6.3 Packet capture

To verify the loss rate we gathered the packets on the sink machine via a program that we developed[1] using the Berkeley Packet Filter [MJ93]. Figure 8 shows that

[1] Not tcpdump. We wrote out our own kernel filter to extract the packets we wanted as well as a user space program to output headers from 2 interfaces simultaneously.

the output of the generator is attached directly to the sink machine as well as the outgoing link of the router. This enables us to capture all the packets and the ones not dropped by the router. A simple difference between the two should verify the loss rate reported by Dummynet. Our `bpf` program captures packets with a specific destination and port, and records the time of arrival, the RTP `src` and `seq` fields.

6.4 Traffic generator

The task of the traffic generator is to create a sequence of packets that resemble many individual IP telephony calls multiplexed together. Furthermore, it should perform this job as accurately as possible with each packet emerging within a given deadline.

Trace file generation and playback In order to be able to subsequently repeat experiments, we first pre-calculate the sending times of the packets and generate trace files. These files are then fed into the traffic generator which sends packets according to the contents. The trace files also allow us to test our setup to see if packets were being generated at the right times, such as inter-arrival times and sequence. The files are generated on a per source basis. The average length of a burst is calculated as shown in Equation 2. An example of a trace file[2] with ten sources is shown in Figure 10.

$$burst\ length = \text{rand}\left(\frac{P_{\text{on}}}{interval}\right) \tag{2}$$

The C-code for the rand function is shown in Figure 9.

```
#define INVERSE_M ((double)   4.6566128200e-10) /* little number */

int calc_length(double burstlen) {
  double rand, logvalue;

  rand = INVERSE_M * random();
  logvalue = burstlen * -log(rand);

  return ((int)(logvalue + 0.5));
}
```

Fig. 9. C code to ‘‘randomize’’ a burst length

Using the logarithm of a uniform random variable generates burst lengths which are exponentially distributed. The same calculation is applied for the idle (with P_{off}) periods. The result is (reading vertically for each source) an exponentially distributed series of ON and OFF sequences with a mean ON of 0.351 seconds, OFF of 0.65 seconds which results in a burst length of 22 packets.

[2] Actually it is converted into a binary format for more compact representation.

```
            source
     0 1 2 3 4 5 6 7 8 9
time
0    0 1 0 1 0 1 0 1 0 1
1    0 1 1 0 1 1 0 1 1 1
2    1 1 1 1 1 1 0 1 0 1
3    1 0 1 0 1 0 1 0 0 1
4    1 0 1 0 1 0 1 0 0 1
5    0 1 1 0 1 1 0 1 1 0
```

Fig. 10. Traffic generator trace file.

The file shows for each time step (in this case 16 ms) which of the 10 sources are on or off. In the example, sources 1, 3, 5, 7 and 9 sends packets in the first time step.

If there are n sources, each timestep is further subdivided into n sub steps. Each sub step defines the sending interval for each source. For example, with ten sources and a time step of 16 ms starting at t, source 0 sends its packet within $[t, t+1.6]$; source 1 sends within $[t+1.6, t+3.2]$, etc. If a source does not send its packets within its interval, it is said to miss its deadline. Packets that miss their deadline are recorded by the generator and recorded. Also the largest value by which a packet was delayed is kept.

For the trace file above, the first steps of a packet sequence is shown in Figure 11. In the figure, the packets of source 5 and 7 missed their deadlines.

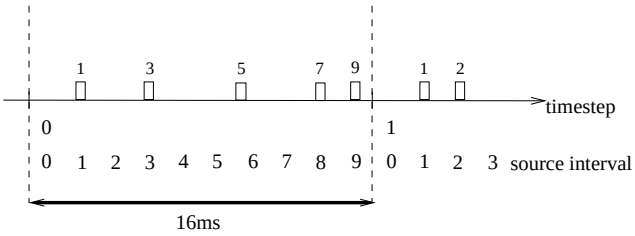

Fig. 11. Traffic generator sending times

The actual sending time on the link can be measured by an external mechanism, such as the packet capture program described previously.

Traffic generator verification As a simple test for a trace file of 220000 packets we obtained values 36.9 % for the ON time, 63.1 % for the OFF time by simply counting the ones and zeros in one column of the file. The mean number of packets in a burst equalled 22.5. Using the trace files turned out to be more useful than we first expected, despite the performance gains of replaying pre-calculated files they also allowed us to test the performance of our traffic

generator (setting all the sources on), cross check parameters as just stated and generating highly correlated sequences for analysing the queue behaviour.

Traffic generator verification We calculated the index of dispersion of intervals for the lab traffic generator. In Figure 12 we can see that the simulation

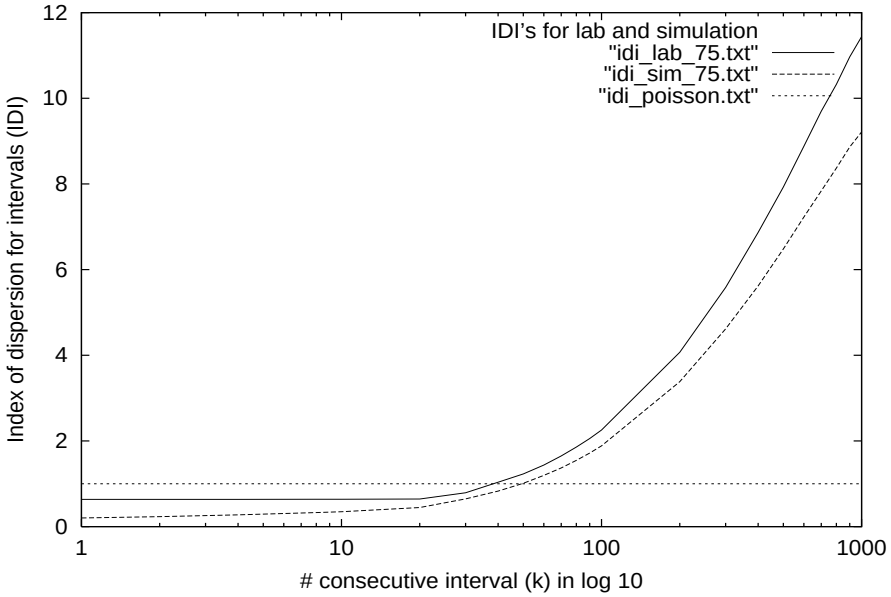

Fig. 12. IDIs for the superposition of 75 sources

and lab traffic generator produce similar amounts of burstiness. The larger the observation time the more skewed the traffic is. One voice source has a burstiness of about 18.1. The plot shows the result of a trace which was 10000 simulated seconds, resulting in 17.3 million packets for the lab and 16.3 for the simulation. A second purpose of the test was to confirm that the traffic generator (and host machine) were capable of transmitting packets as close as possible to their deadlines. When making comparative studies it is meaningful, at the onset, to strive for reducing any unknowns in the input data.

7 Results

In this section we present and discuss the results from the MMPP model, the `ns-2` simulations and the measurements from the lab setup. Recall from Section 4 that in all three cases we used 32 kb/s ADPCM voice encoding with 16 ms packetization. This results in 64 bytes of voice payload in each packet and a total packet size of 104 bytes including the RTP, UDP and IP protocol headers.

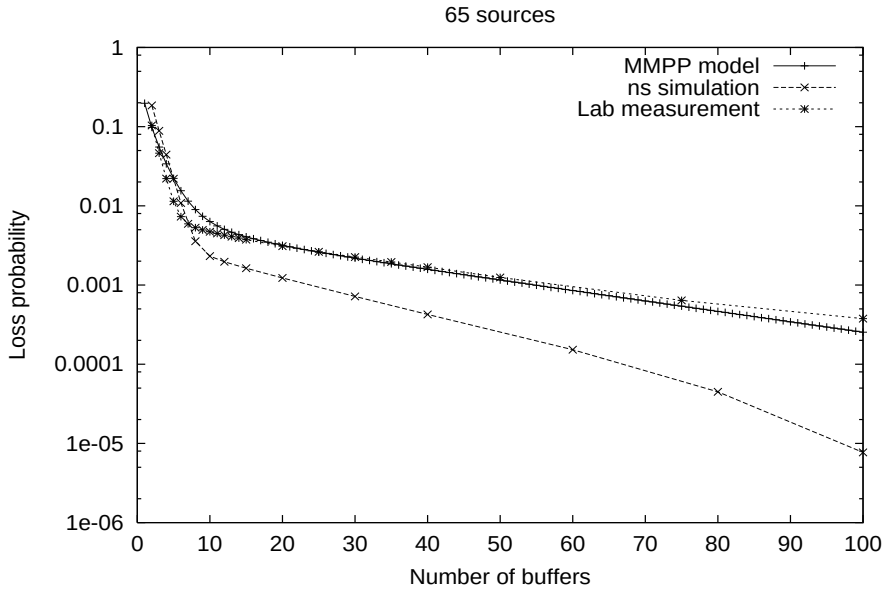

Fig. 13. 65 sources for model, `ns-2` and lab

Figures 13 and 14 show the packet loss probability as a function of the number of buffers on a y-log scale. We can see in these graphs that both the MMPP model results and the `ns-2` simulations, in general, are comparable with the measurements in the lab. The exception is for large buffer sizes where the loss probabilities become low.

The MMPP model in this case is closer to the lab measurements than the `ns-2` simulations. The `ns-2` simulations consistently show the lowest loss rates for more than 7–8 buffers. We analysed the output from the traffic generators in `ns-2` and in the lab in order to find an explanation. We found that there is a small difference in the mean total rate between them which could be one explanation.

The second set of graphs presented in Figures 15 to 18 plot the packet loss probability as a function of the number of voice sources for four different buffer lengths measured in packets. These buffer lengths correspond to a maximum queueing delay of 1.6, 2.7, 5.4 and 21.7 ms, respectively. For a the lower buffer sizes we see saturation effects. In the case of more than 10 buffer places, the lab measurements often shows the highest loss rates. Below about 10 buffers, the lab measurements have the lowest loss rate. From our investigations this was not easy to explain systematically.

One obvious difference in the setups is that the bandwidth offered by Dummynet is not exactly the same as in `ns-2`. Using `netperf` we found there to be about a 3% difference between what `netperf` and dummynet report as their measured and configured bandwidths respectively. Perhaps more subtle and not

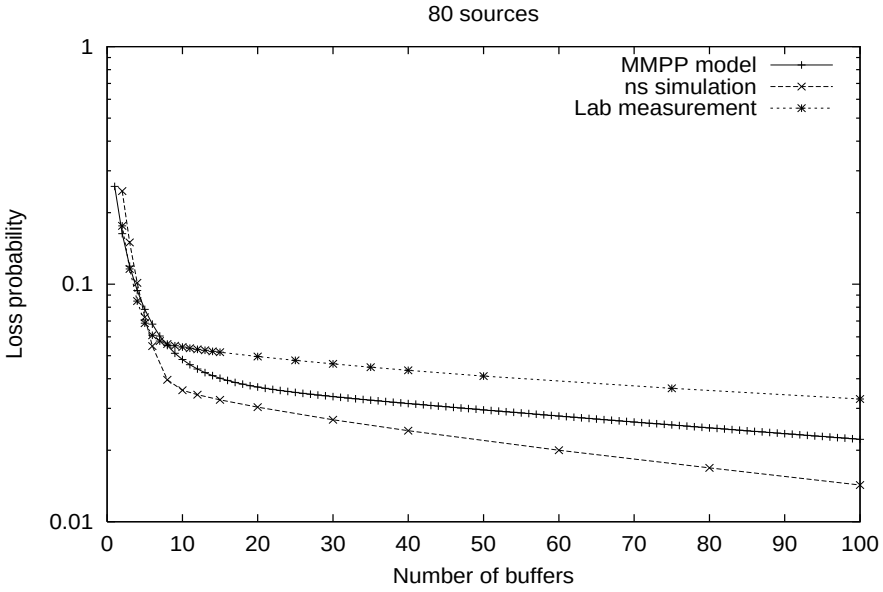

Fig. 14. 80 sources for model, `ns-2` and lab (log scale)

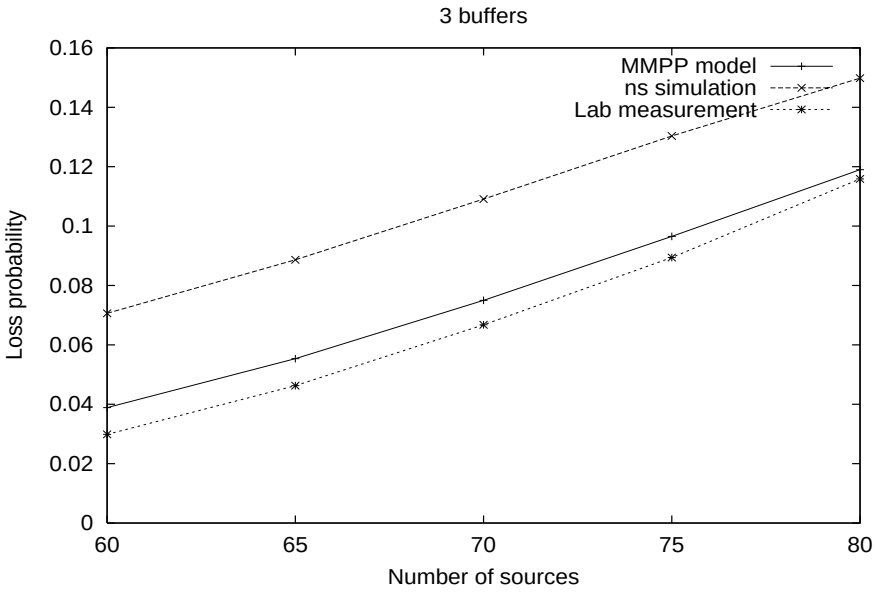

Fig. 15. Loss probability Vs buffers (3)

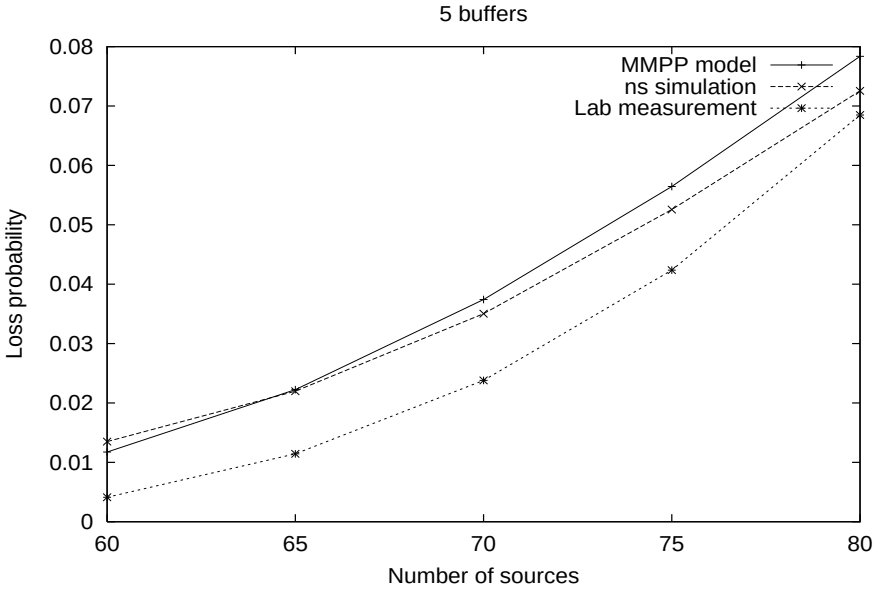

Fig. 16. Loss probability Vs buffers (5)

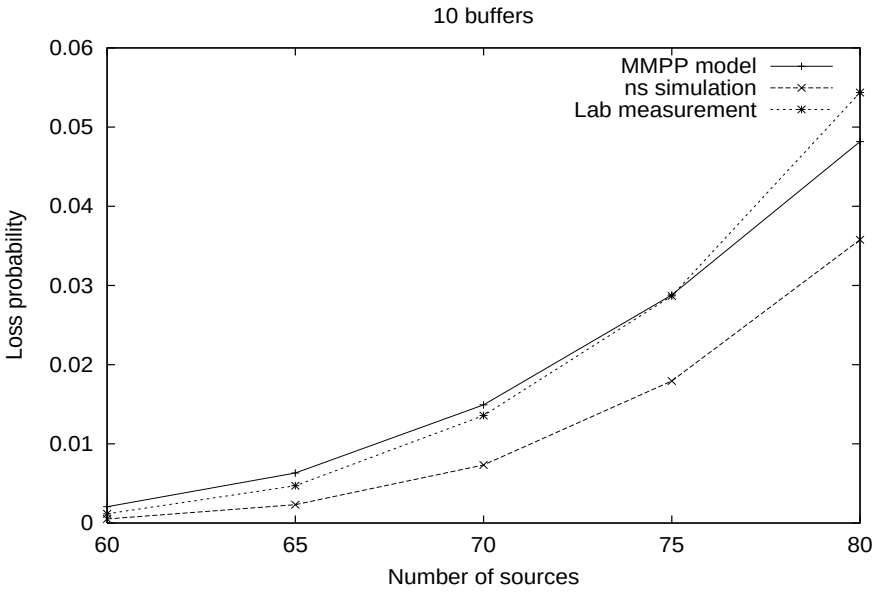

Fig. 17. Loss probability Vs buffers (10)

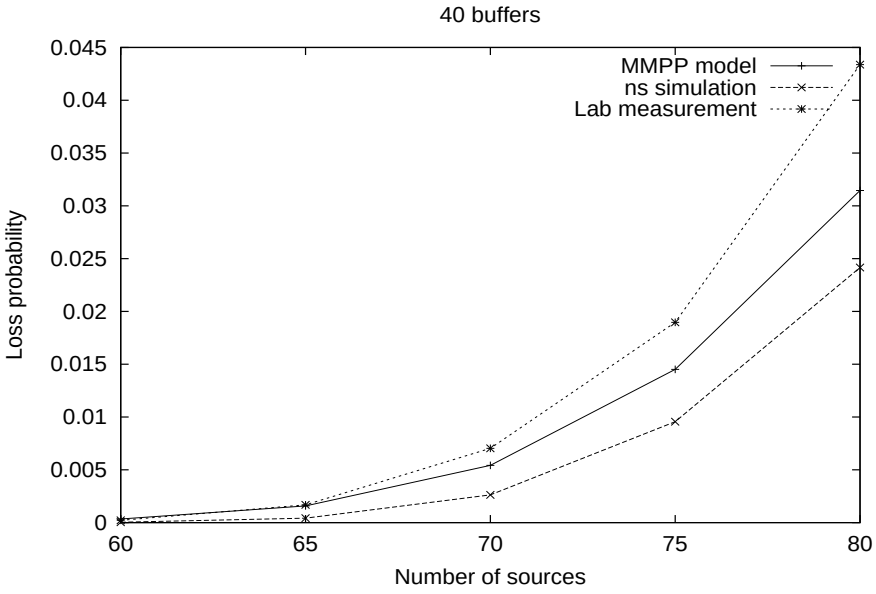

Fig. 18. Loss probability Vs buffers (40)

so obvious is the amount of buffering in the system, in `ns-2` we simply state the buffer size in packets (between 2 and 100). In a real system this is much harder to calculate as buffers exist in many places in the system, for example in the queue between the Ethernet driver and the `ip_input()` routine on the input side. Ethernet cards can also buffer packets on the output side. Nevertheless, the buffering in a real system is probably larger than the simulation and possibly accounts for the differences in the systems under comparison. Additionally as we have seen, there can be differences in the IDI's between the `ns-2` and lab setups.

8 Conclusions and future work

We have studied the packet loss behaviour when a number of homogeneous voice sources are multiplexed onto a bottleneck link. The goal is to find an accurate mathematical model which can be used to dimension a link.

We have implemented a mathematical model based on a Markov modulated Poisson process (MMPP) in Matlab. The model was compared with both simulations using `ns-2` and measurements in a lab environment. The comparison shows that the model in general predicts the loss rate well. An interesting result is that most of the time the model predicts the loss rate better than the `ns-2` simulations.

This result once more proves that the only way to reliably verify a model is to make measurements of a real system. We found that the relationship between

the load and loss rate is close to linear for few buffers (around three), but is exponential for many (10 and above) buffers.

The general conclusion is that the MMPP-based model is pretty well suited for predicting loss rates for superpositioned voice sources in a system with limited buffer space. The mathematical model is an important tool for conveniently dimensioning network links. The lab environment is constrained to physical limits as well as finite resources where the model is clearly not. Running a lab experiment consumes resources and time a lab experiment takes on average 12 hours to complete. The simulation typically takes 2 hours whereas the model consumes only about 10 minutes as well as considerably less physical resources[3].

There are a number of further work items that can be addressed. The maximum delay is bounded by the buffer length in this work, but what is the resulting mean delay? One challenge is to accurately generate enough sources. The next step is to measure a system which has multiple traffic classes in the style of diffserv [NJZ97]. How do different queue scheduling algorithms affect the dimensioning of traffic classes? Can the MMPP model presented in this paper be used to describe the loss and delay properties of other traffic types?

Acknowledgements

The authors would like to acknowledge the indispensable work of Henrik Abrahamsson in helping us calculating the IDI values presented in this paper. We would like thank Thiemo Voigt for his help in setting up Dummynet and adding extra debug statements to make our loss calculations considerably simpler. Finally we would like to thank Telia AB for their financial support in the early phase of this work.

References

[AMS82] D. Anick, Debasis Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal*, 61(8):1871–1894, October 1982.

[And00] Anders Andersson. Capacity study of statistical multiplexing for ip telephony. Technical Report T2000:03, SICS – Swedish Institute of Computer Science, January 2000.

[BBMR91] Andrea Baiocchi, Nicola Blefari-Melazzi, and Aldo Roveri. Buffer dimensioning criteria for an ATM multiplexer loaded with homogeneous on-off sources. In J. W. Cohen and Charles D. Pack, editors, *Queueing, Performance and Control in ATM — Proceedings of the Workshop at the 13th International Teletraffic Congress (ITC)*, pages 13–18, Copenhagen, Denmark, June 1991. North-Holland. Volume 15 of the North Holland Studies in Telecommunication.

[3] These values were derived from an Athlon 600 Mhz PC with FreeBSD, a Fast SCSI-3 disk and 128 MB of RAM.

[BML+91] Andrea Baiocchi, Nicola Blefari Melazzi, Marco Listanti, Aldo Roveri, and Roberto Winkler. Loss performance analysis of an ATM multiplexer loaded with high-speed on-off sources. *IEEE Journal on Selected Areas in Communications*, 9(3):388–393, April 1991.

[FV98] Kevin Fall and Kannan Varadhan. ns: Notes and documentation. Technical report, Berkeley University, 1998.

[Gut95] Allan Gut. *An Intermediate Course in Probability.* Springer-Verlag, New York, 1995.

[HL86] Harry Heffes and David M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):856–867, September 1986.

[Int93] International Telecommunication Union (ITU). Transmission systems and media, general recommendation on the transmission quality for an entire international telephone connection; one-way transmission time. Recommendation G.114, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, March 1993.

[MJ93] Steven McCanne and Van Jacobson. A BSD packet filter: A new architecture for user-level packet capture. In *Proc. of Usenix Winter Conference*, pages 259–269, San Diego, California, January 1993.

[NJZ97] Kathie Nichols, Van Jacobson, and Lixia Zhang. A two-bit differentiated services architecture for the internet. Internet draft, Bay Networks, LBNL and UCLA, November 1997.

[NKT91] Ramesh Nagarajan, James F. Kurose, and Don Towsley. Approximation techniques for computing packet loss in finite-buffered voice multiplexers. *IEEE Journal on Selected Areas in Communications*, 9(3):368–377, April 1991.

[Riz97] L. Rizzo. Dummynet: A simple approach to the evaluation of network protocols. *Computer Communications Review*, 27(1):31–41, January 1997.

[Sun98] Zheng Sun. Capacity study of statistical multiplexing for ip telephony. Technical Report LiTH-MAT-EX-98-12, Department of Mathematics, Linkoping University, December 1998.

[SW86] Kotikalapudi Sriram and Ward Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):833–846, September 1986.

[Tuc88] Roger C. F. Tucker. Accurate method for analysis of a packet-speech multiplexer with limited delay. *IEEE Transactions on Communications*, COM-36(4):479–483, April 1988.

Paper B

Ingemar Kaj and Ian Marsh.
Modelling the Arrival Process for Packet Audio. In *Quality of Service in Multiservice IP Networks*, pages 35-49, Milan, Italy, February 2003.

"As my anger shouts at my own self doubts,
so sadness creeps into my dreams"

Paul Weller - Above the clouds

Modelling the Arrival Process for Packet Audio

Ingemar Kaj[1] and Ian Marsh[2]

[1] Dept. of Mathematics, Uppsala University, Sweden
ikaj@math.uu.se
[2] SICS AB, Stockholm, Sweden
ianm@sics.se

Abstract. Packets in an audio stream can be distorted relative to one another during the traversal of a packet switched network. This distortion can be mainly attributed to queues in routers between the source and the destination. The queues can consist of packets either from our own flow, or from other competing flows. The contribution of this work is a Markov model for the time delay variation of packet audio on the Internet. Our model is extensible, and show this by including sender silence suppression and packet loss into the model. By comparing the model to wide area traffic measurements we show the possibility to generate an audio arrival process similar to those created by operating conditions.

1 Introduction

Modelling the arrival process for audio packets that have passed through a series of routers is the problem we will address. Figure 1 illustrates this situation: Packets containing audio samples are sent at a constant rate from a sender, shown as step one. The spacing between packets is compressed and elongated

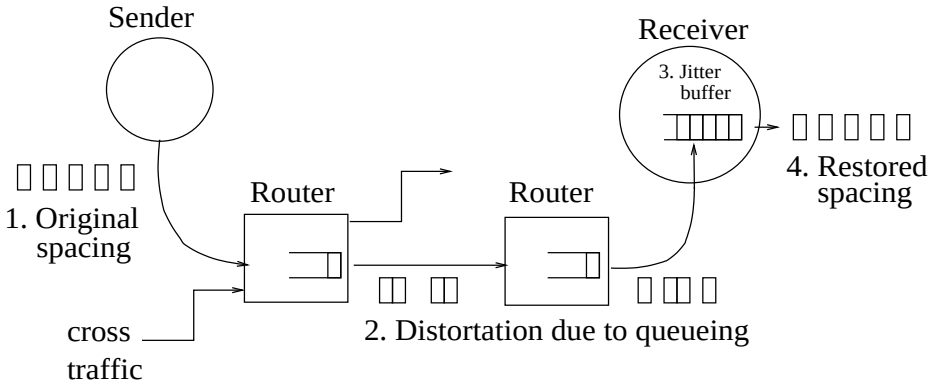

Fig. 1. The networks effect on packet audio spacing

relative to each other is due to the buffering in intermediate routers and mixing

with cross-traffic, shown as step two. In order to replay the packets with their original spacing, a buffer is introduced at the receiver, commonly referred to as a *jitter buffer* shown as step three. The objective of the buffer is to absorb the variance in the inter-packet spacing introduced by the delays due to cross traffic, and (potentially) its own data. In step four, using information coded into the header of each packet, the packets are replayed with their original timings restored.

The motivation for this work derives from the inability of using known arrival processes to approximate the packet arrival process at the receiver. Using a known arrival process, even a complex one, is not always realistic as the model does not include characteristics that real audio streams experience. For example the use of silence suppression or the delay/jitter contribution of cross traffic. One alternative is to use real traffic measurements. Although they produce accurate and representative arrival processes, they are inherently static and do not offer much in the way of flexibility. For example, it is impossible to observe the effect of using different packet sizes without redoing all the experiments. When testing the performance of jitter buffer playout algorithms, for example, this inflexibility is undesirable. Thus, an important contribution of this paper is to address the deficiencies of these approaches by *combining* the advantages of both a model of the process, with using data from real measurements.

This paper presents in a descriptive manner, a packet delay model, based on the main assumption that packets are subjected to independent transmission delays. It is intended that readers not completely familiar with Markovian theory can follow the description. We assume no prior knowledge of the model as it is built from first principles starting in section 2. We give results for the mean arrival and interarrival times of audio packets in this section also. We add silence suppression to the model in section 3 and packet loss in section, 4. Real data is incorporated in section 5, related work follows in section 6 and we customarily round off with some conclusions in section 7.

2 The packet delay model

There are two causes of delay for packet audio streams. First, the delay contributed by cross traffic, usually TCP traffic, which we will call the *transmission* delay in this paper. Second, the delay caused by our own packets, i.e. those queue up behind ones from the same flow, this we refer to as the *sequential* delay. It is important to state we consider these two delays as separate, but study their combined interaction on the observed delays and interarrivals. Propagation and scheduling delay are not modelled as part of this work.

In this model packets are transmitted periodically using a packetisation time of 20 milliseconds. For convenience, the packetisation interval is used as the time unit for the model. Saying that a packet is sent at time k signifies that this particular packet is sent at clock time $20k$ ms into the data stream. The first packet is sent at time 0.

We begin with the transmission delay of a packet. Suppose that packet k could be sent isolated from the rest of the audio stream and let

$$Y_k = \text{transmission delay of packet } k \text{ (no. of 20 ms periods)}.$$

To see the impact of the sequential delay, let

$$T_k = \text{the arrival time of packet } k \text{ at the jitter buffer}, \quad k \geq 1.$$

The model used in this paper is shown in Figure 2. The figure shows packets

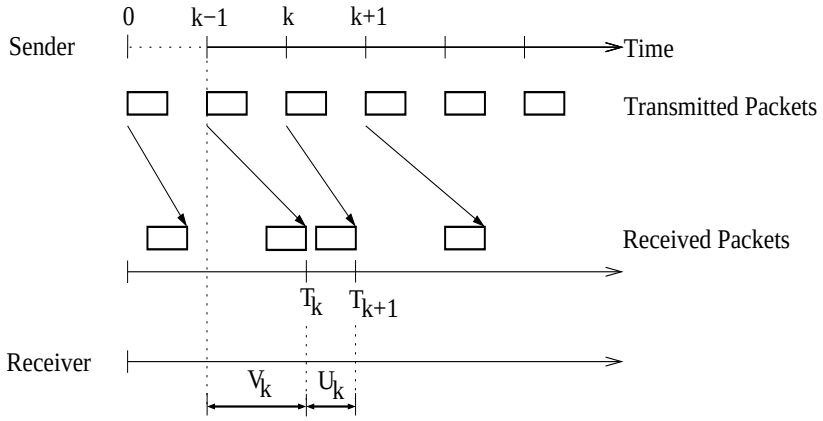

Fig. 2. T_k arrival times before playout, V_k observed delays, U_k observed interarrival times

being transmitted from a sender at regular intervals. They traverse the network, where as stated, their original spacing is distorted. Packet k arrives at time T_k at the receiver. The difference in time between when it departed and arrived we call the *observed delay*, which we denote

$$V_k = \text{arrival time} - \text{departure time} = T_k - k + 1 \quad k \geq 1.$$

The time when the next packet (numbered $k+1$) arrives is T_{k+1} and so the *observed interarrival times* are obtained as the differences between T_{k+1} and T_k, denoted

$$U_k = T_{k+1} - T_k.$$

A packet k, sent at time $k-1$, requires time Y_k to propagate through the network and arrives therefore at $T_k = k - 1 + Y_k$. It may however catch up to audio packets transmitted earlier ($1 \rightarrow k-1$) which we called sequential delays. This packet is forced to wait before arriving to the receiver's playout buffer. This shows that the actual arrival times satisfy:

$$\begin{aligned} T_1 &= Y_1 \\ T_k &= \max(T_{k-1},\, k - 1 + Y_k), \quad k \geq 2. \end{aligned} \tag{1}$$

Since T_{k-1} and Y_k are independent, we conclude from the relation above (1) that T_k forms a transient Markov chain. Moreover, the interarrival times satisfy

$$U_k = T_k - T_{k-1} = \max(0,\, k - 1 + Y_k - T_{k-1}) \quad k \geq 2. \tag{2}$$

The arrival times (T_k), interarrival times (U_k) and observed delays (V_k) can be easily observed from traffic measurements. As an example, Figure 3 shows the histogram for an empirical sequence of interarrival times. The data is from a

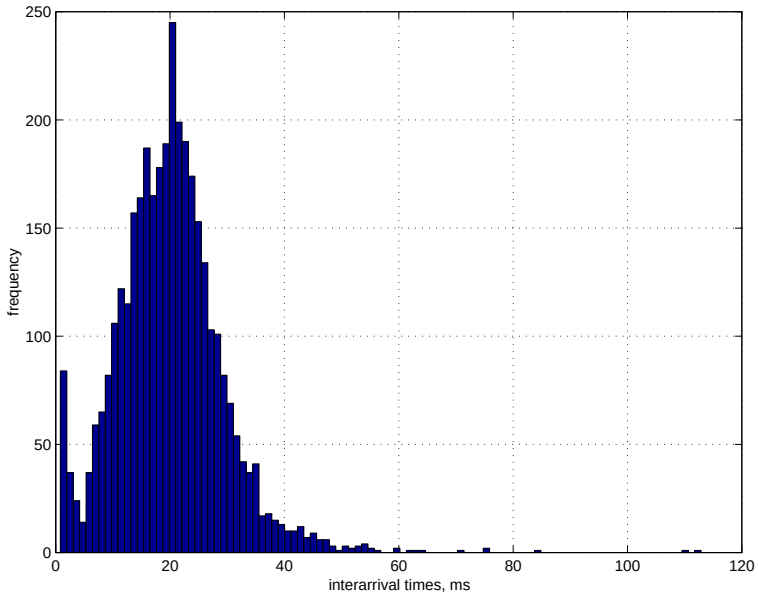

Fig. 3. Histogram of the interarrival times (U_k)

recording of a Voice over IP session between Argentina and Sweden, more details of the traffic measurements are given later in section 5.1. The transmission delay sequence (Y_k) should be on the other hand considered as non-observable. The approach in this study is to consider (Y_k) having a general (unknown) distribution and investigate the resulting properties of the observed delay (V_k) and interarrival times (U_k). Since the latter sequences can be empirically observed, this leads to the question to whether the transmission delay distribution can be reconstructed using statistical inference. In this direction we will indicate some methods that have been used to compare the theoretical results with the gathered empirical data.

To carry out the analysis, we assume from this point the sequence (Y_k) is independent and identically distributed with distribution function

$$F(x) = P(Y_k \leq x), \quad k \geq 1,$$

and finite mean transmission delay $\nu = \int_0^\infty (1 - F(x))\, dx < \infty$. For the data in our study, typical values of ν are 20-40, i.e. 400-800 ms. We consider the above assumptions justified for the purpose of studying a reference model, obviously it would be desirable to allow dependence over time.

2.1 Mean arrival and interarrival times

It is intuitively clear that in the long run $E(U_k) \approx 1$ as on average packets arrive with 20 ms spacing, which we will now verify for the model. The representation (1) for T_k can be written

$$T_k = \max(Y_1, 1 + Y_2, \ldots, k - 1 + Y_k) \quad k \geq 1,$$

which gives the alternative representation

$$T_k = \max(Y_1, 1 + T'_{k-1}), \quad k \geq 2 \tag{3}$$

where on the right side

$$T'_{k-1} = \max(Y_2, 1 + Y_3, \ldots, k - 2 + Y_k)$$

has the same marginal distribution as T_{k-1} but is independent of Y_1. From (3) it follows that we can write $\{T_k > t\}$ as a union of two disjoint events, as

$$\{T_k > t\} = \{1 + T'_{k-1} > t\} \cup \{Y_1 > t, 1 + T'_{k-1} \leq t\}.$$

Hence, using the independence of T'_{k-1} and Y_1,

$$\begin{aligned} P(T_k > t) &= P(1 + T'_{k-1} > t) + P(Y_1 > t, 1 + T'_{k-1} \leq t) \\ &= P(1 + T_{k-1} > t) + P(Y_1 > t)P(1 + T_{k-1} \leq t) \end{aligned}$$

and so

$$\begin{aligned} E(T_k) &= \int_0^\infty P(T_k > t)\, dt \\ &= E(1 + T_{k-1}) + \int_1^\infty P(Y_1 > t)P(T_{k-1} \leq t - 1)\, dt. \end{aligned} \tag{4}$$

Therefore

$$E(U_k) = 1 + \int_1^\infty P(Y_1 > t)P(T_{k-1} \leq t - 1)\, dt \to 1, \quad k \to \infty \tag{5}$$

(since $\nu = \int_0^\infty P(Y_1 > t)\, dt < \infty$ and $T_k \to \infty$, the dominated convergence theorem applies forcing the integral to vanish in the limit).

A further consequence of (4) is obtained by iteration,

$$E(T_k) = k - 1 + E(Y_1) + \int_1^\infty P(Y_1 > t) \sum_{i=1}^{k-1} P(T_i \leq t - 1)\, dt.$$

If we introduce

$$N(t) = \text{the number of arriving packets in the time interval } (0, t],$$

so that $\{N(t) \geq n\} = \{T_n \leq t\}$, this can be written

$$E(V_k) = E(Y_1) + \int_1^\infty P(Y_1 > t) \sum_{i=1}^{k-1} P(N(t-1) \geq i)\, dt, \tag{6}$$

which, as $k \to \infty$, gives an asymptotic representation for the average observed delay as

$$E(V_k) \to \nu + \int_1^\infty P(Y_1 > t) E(N(t-1)))\, dt. \tag{7}$$

2.2 Steady state distributions

By (1),

$$P(T_k \leq x) = \prod_{i=1}^{k} P(i + Y_i \leq x + 1) = \prod_{i=0}^{k-1} F(x - i),$$

and therefore the sequence (V_k), which we defined by $V_k = T_k - k + 1$, $k \geq 1$, satisfies

$$P(V_k \leq x) = \prod_{i=0}^{k-1} F(x + k - 1 - i) = \prod_{i=0}^{k-1} F(x + i) \quad x \geq 0.$$

This shows that (V_k) is a Markov chain with state space the positive real line and asymptotic distribution given by

$$P(V_\infty \leq x) = \prod_{i=0}^{\infty} F(x + i) \quad x \geq 0. \tag{8}$$

Furthermore, for $x \geq 0$

$$\begin{aligned} P(U_k \geq x) &= P(k - 1 + Y_k - T_{k-1} \geq x) = P(V_{k-1} \leq Y_k + 1 - x) \\ &= \int_0^\infty P(V_{k-1} \leq y + 1 - x)\, dF(y), \end{aligned}$$

where in the step of conditioning over Y_k we use the independence of Y_k and V_{k-1}. Therefore the sequence (U_k) has the asymptotic distribution

$$P(U_\infty \leq x) = 1 - \int_0^\infty \prod_{i=1}^{\infty} F(y - x + i)\, dF(y) \quad x \geq 0, \tag{9}$$

in particular a point mass in zero of size

$$P(U_\infty = 0) = 1 - \int_0^\infty \prod_{i=1}^{\infty} F(y + i)\, dF(y). \tag{10}$$

This distribution has the property that $E(U_\infty) = 1$ for any given distribution F of Y with $\nu = E(Y) < \infty$. In fact, this follows from 5 under a slightly stronger assumption on Y (uniform integrability), but can also be verified directly by integrating (9).

Figure 4 shows numeric approximations of the (non-normalised) density function $\frac{d}{dx}P(U_\infty \leq x)$ of (9) for three choices of F. All three distributions show a characteristic peak close to time 1 corresponding to the bulk of packets arriving with the default interarrival spacing of 20 ms. A fraction of the probability mass is fixed at $x = 0$ in accordance with (10), but not shown explicitly in the figure. These features of the density functions can be compared with the shape of the histogram in Figure 3 with its peak at the 20 ms spacing. Also, close to the origin there is a small peak which corresponds to packets arriving back-to-back usually arriving as a burst, possibly due to a delayed packet ahead of them. In Figure 4 the density function with the highest peak close to 1 time unit is a Gaussian distribution with arbitrarily selected parameters mean 5 and variance 0.2. Of the two exponential distributions, the one with the higher variance (Exp(3)) has a lower peak and more mass at zero compared to an exponential with a smaller variance (Exp(2)).

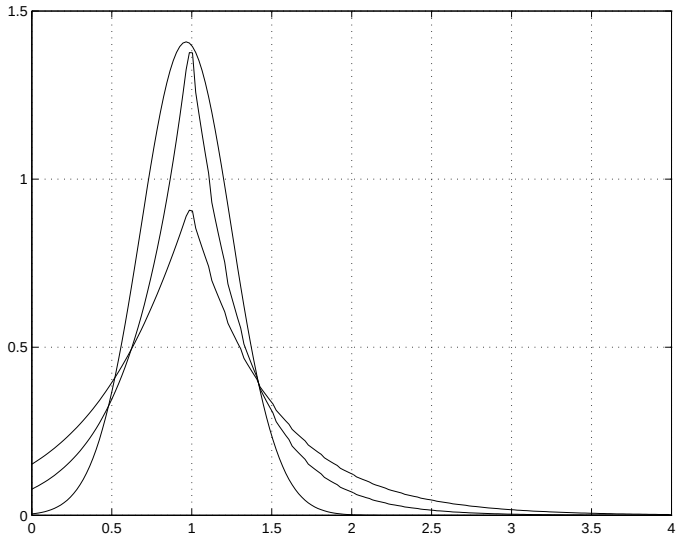

Fig. 4. Density functions of U for N(5,0.2), Exp(2) and Exp(3)

3 Silence suppression mechanism

In this section we incorporate into the model an additional source of random delays due to speaker silence suppression. Silence suppression is employed at the

sender so as not to transmit packets when there is no speech activity. During a normal conversation this accounts for about half of the total number of packets, thus considerably reducing the load on the network. Assign to packet number k the quantity

$$X_k = \text{the silence period duration between packets } k-1 \text{ and } k.$$

We assume that the silence suppression intervals are independent of $(Y_k)_{k\geq 1}$ and are given by a sequence of independent random variables $X_1, X_2, \ldots$, such that

$$G(x) = P(X_k \leq x), \quad 1-\alpha = G(0) = P(X_k = 0) > 0, \quad \mu = E(X_k) < \infty.$$

The (small) probability $\alpha = P(X_k > 0)$ represents the case where silence suppression is activated just after packet $k-1$ is transmitted from the sender. Note that

$$S_k = \sum_{i=1}^{k} X_i = \text{total time of silence suppression affecting packet } k,$$

which implies that the delivery of packet k from the sender now starts at time $k-1+S_k$. The representation (1) takes the form

$$T_1 = S_1 + Y_1, \quad T_k = \max(T_{k-1},\, k-1+S_k+Y_k), \quad k \geq 2, \tag{11}$$

hence

$$U_k = T_k - T_{k-1} = \max(0,\, k-1+S_k+Y_k-T_{k-1}) \quad k \geq 2. \tag{12}$$

Similarly,

$$V_k = \text{arrival time} - \text{departure time} = T_k - k + 1 - S_k \quad k \geq 1.$$

The alternative representation of (3) is

$$T_k = X_1 + \max(Y_1, 1 + T'_{k-1}), \tag{13}$$

where

$$T'_{k-1} = \max(Y_2 + S_2 - X_1, 1 + Y_2 + S_2 - X_1, \ldots, k-2+Y_k+S_k-X_1)$$

has the same marginal distribution as T_{k-1} but is independent of X_1 and Y_1. In analogy with the calculation of the previous section leading up to (4), this relation gives

$$E(T_k) = E(X_1 + 1 + T_{k-1}) + \int_1^\infty P(X_1 + Y_1 > t, X_1 + T'_{k-1} \leq t-1)\, dt. \tag{14}$$

Exchanging the operations of integration and expectation shows that the previous integral can be written

$$E\left[\int_{1+X_1}^{\infty} \mathbf{1}\{Y_1 > t - X_1, T'_{k-1} > t - X_1 - 1\}\, dt\right]$$

where we have also used that the integrand vanishes on the set $\{t \le 1+X_1\}$. Apply the change-of-variables $t \to t - X_1$ to get $E\left[\int_1^\infty \mathbf{1}\{Y_1 > t, T'_{k-1} > t-1\}\, dt\right]$. Then shift integration and expectation again to obtain from (14) the relations

$$E(T_k) = 1 + E(X_1) + E(T_{k-1}) + \int_0^\infty P(Y_1 > t)P(T_{k-1} \le t-1)\, dt$$

and

$$E(U_k) = 1 + E(X_1) + \int_1^\infty P(Y_1 > t)P(T_{k-1} \le t-1)\, dt.$$

Hence with silence suppression, as $k \to \infty$,

$$E(U_k) \to 1 + \mu, \quad E(V_k) \to \nu + \int_1^\infty P(Y_1 > t)E(N(t-1))\, dt, \qquad (15)$$

using the same arguments as in the simpler case of the previous section.

4 Including packet loss in the model

We return to the original model without silence suppression but consider instead the effect of lost packets. Suppose that each IP packet is subject to loss with probability p, independently of other packet losses and of the transmission delays. Lost packets are unaccounted for at the receiver and hence the sequence (T_k) records the arrival times of received packets only. To keep track of the delivery times of sent and received packets introduce

$$\begin{aligned} K_k = &\text{ number of attempts required between} \\ &\text{successfully received packets } k-1 \text{ and } k, \quad k \ge 1, \end{aligned}$$

which gives a sequence $(K_k)_{k\ge1}$ of independent, identically distributed random variables with a geometric distribution

$$P(K_k = j) = (1-p)p^j, \quad j \ge 0.$$

Moreover,

$$\begin{aligned} L_k &= K_1 + \ldots + K_k \\ &= \text{number of attempts required for } k \text{ successful packets} \end{aligned}$$

is a sequence of random variables with a negative binomial distribution. The arrival times of packets are now given by

$$T_1 = K_1 - 1 + Y_{K_1}, \quad T_k = \max(T_{k-1},\, L_k - 1 + Y_{L_k}), \quad k \ge 2.$$

Due to the independence we may re-index the sequence of Y_{L_k}'s to obtain

$$T_1 = K_1 - 1 + Y_1, \quad T_k = \max(T_{k-1},\, L_k - 1 + Y_k), \quad k \ge 2.$$

and thus

$$T_k = K_1 - 1 + \max(Y_1, 1 + T'_{k-1}), \quad k \geq 2 \tag{16}$$

with K_1, Y_1 and T'_{k-1} all independent, and again T_{k-1} and T'_{k-1} identically distributed. This is the same relation as (13) with X_1 replaced by $K_1 - 1$ and hence, as in (15), $E(U_k) \to 1 + E(K_1 - 1) = \frac{1}{1-p}$, $k \to \infty$, which provides a simple method to estimate packet loss based on observed interarrival times. Similarly, combining silence suppression and packet loss,

$$E(U_k) \to 1 + E(X) + E(K_1 - 1) = \mu + \frac{1}{1-p}, \quad k \to \infty, \tag{17}$$

5 Incorporating Real Data

5.1 Trace data

We now give a brief description of the experiments we performed in order to obtain estimates for the parameters in the model. Pulse Code Modulated (PCM) packet audio streams were sent from a site in Buenos Aires, Argentina to Stockholm, Sweden over a number of weeks. The streams were sent as a 64kbits/sec rate in 160 byte payloads. This implies the packets leave the sender with a interpacket spacing of 20 ms. The remote site is approximately 12,000 kilometres, 25 Internet hops and four time zones from our receiver. The software was capable of performing silence suppression, in which packets are not sent when the speaker is silent. Without silence suppression, 3563 packets were sent during 70 seconds and with suppression 2064 were sent. We record the absolute times the packets leave the sender and the absolute arrival times at the receiver. This gives an observed sequence

$$v_k = \text{arrival time no } k - \text{departure time no } k$$

of the Markov chain (V_k). In particular, the sample mean $\bar{v}$ is an estimate of the one-way delay. Similarly,

$$u_k = \text{arrival time no } k - \text{arrival time no } (k-1)$$

is a sample of the interarrival time sequence (U_k).

A typical sequence of measurement data used in this study *without* silence suppression is shown in Figure 5, which shows (v_k) and (u_k) for a small sequence of 200 packets ($1700 \leq k < 1900$), corresponding to four seconds of speech. To further illustrate such measurement data, Figure 6 shows a histogram of the delays (v_k) and Figure 3 showed a histogram of the interarrival times (u_k). It can be noted that large values of interarrival times are sometimes followed by very small ones, manifesting that a severely delayed packet forces subsequent packets to arrive back-to-back. The fraction of packets arriving in this manner corresponds to the height of the leftmost peak in the histogram of Figure 3.

Returning to measurements with silence suppression, Figure 7 shows the statistics of the voice signal used. The upper histogram shows the talkspurts

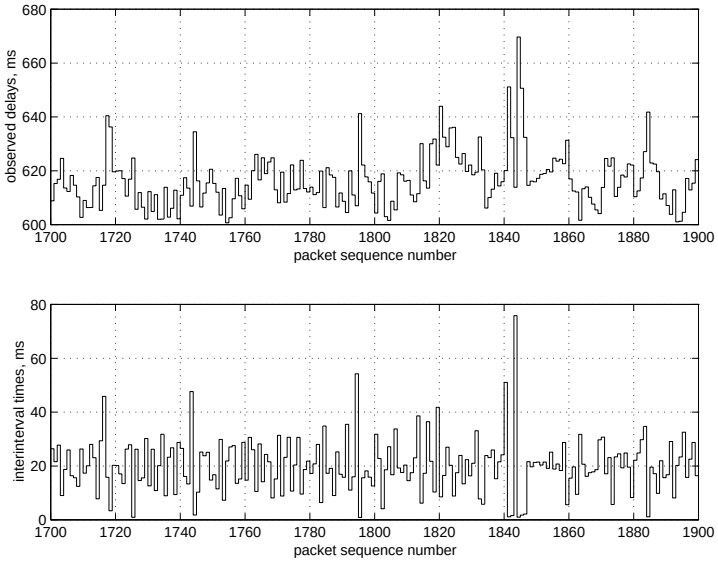

Fig. 5. Four second speech packet measurements: a)delays b)interarrival times

and the lower histogram the non-zero part of the distribution G of the silence intervals X. The probability $\alpha = P(X = 0)$ and the expected value $\mu = E(X)$ were estimated as

$$\alpha^* = 0.0456 \qquad \mu^* = 25.7171.$$

5.2 Numerical estimates

In this section we indicate a few simple numerical techniques that give parameter estimates based on measurement data. In principle such methods based on the model presented here can be used for systematic studies of delays and losses for comparison of measurements sampled in different environments.

Considering first the case of no silence suppression, it was pointed out in section 4 that given an observed realization $(u_k)_{k=1}^n$ of (U_k), a point estimate of the packet loss probability p is obtained from (17) (with $\mu = 0$), using

$$p^* = 1 - \frac{20}{\bar{u}}, \qquad \bar{u} = \frac{1}{n}\sum_{k=1}^{n} u_k \text{ ms}.$$

Our measurements gave consistently $\bar{u} \approx 20.002 - 20.005$ ms, indicating loss probabilities in the order of 10^{-4}.

Next we look at an experiment where the pre-recorded voice is transmitted at seven different times using silence suppression, and look at the interarrival times measured at the receiver during each transmission. Table 1 shows the expected

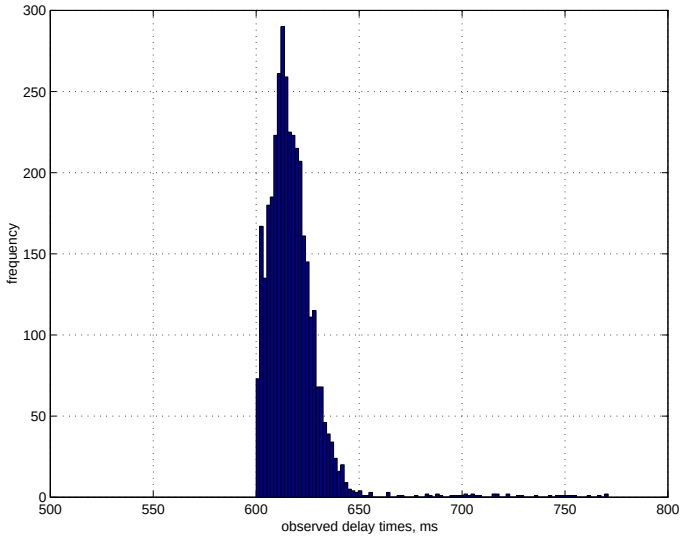

Fig. 6. Histogram of the observed delays (V_k)

Table 1. Silence Interval Parameters

Trace	**E(X)**	**E(X)-μ^***
trace 1	25.7492	0.0321
trace 2	26.2204	0.4639
trace 3	26.2284	0.5113
trace 4	26.2164	0.4993
trace 5	26.2186	0.5015
trace 6	26.2124	0.4953
trace 7	26.2209	0.5038

silence interval $E(X)$ and the estimated μ from the measurement data. The obtained estimates indicate a systematic bias in the order of 0.5 milliseconds in the mean values of the silence suppression intervals. Packet losses cannot fully explain the observed deviation, however for the present preliminary investigation we find the numerical estimates satisfactory. A more comprehensive statistical analysis might reveal the source of this slight mismatch.

We now consider the problem of estimating the distribution F of packet delays Y given a fixed length sample observation (v_k) of the Markov chain (V_k) for observed delays. One method to do this is to base it on the steady state analysis already presented in section 2.2. Indeed, rewriting (8) as the simple relation

$$P(V_\infty \leq x) = F(x) \prod_{i=1}^{\infty} F(x+i) = F(x)\, P(V_\infty \leq x+1)$$

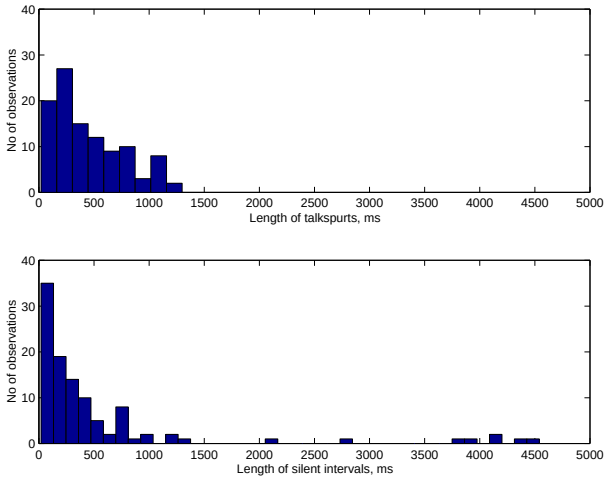

Fig. 7. Lengths of talkspurts and silence periods

shows that if we let $\bar{F}_V$ denote an empirical distribution function of V, then we obtain an estimate $\bar{F}$ of F by taking

$$\bar{F}(x) = \frac{\bar{F}_V(x)}{\bar{F}_V(x+1)} \quad x \geq 0, \tag{18}$$

where we recall that the variable x is measured in units of 20 ms intervals. An application of this numerical algorithm to the measurement data of the previous figures (5 and 6) yields an estimated density function for Y as in Figure 8. The numerical scheme is sensitive for small changes in the data, so it is difficult to draw conclusions on the finer details of the distribution of F. As expected the graph is very similar to that of the observed delays shown in Figure 6, but with certain differences due to the Markovian dependence structure in the sequence (V_k) as opposed to the independence in (Y_k). The main difference is the shift towards smaller values for Y in comparison to those of V. This corresponds to the inequality $\bar{F}(x) \geq \bar{F}_V(x)$ valid for all x, which can be seen from (18).

6 Related Work

Many researchers have looked at the needs in terms of buffer size for packet streams characterised by Markov (semi or modulated) behaviour especially in the case of multiplexed sources. Their goal was to derive the waiting time of packets spent in the buffer shown as a probability density function of the waiting times. Relatively few, however, have looked at the arrival process using a stage of buffers and identifying embedded Markov chains from a single source. We concentrated on this scenario, including both streams with and without silence

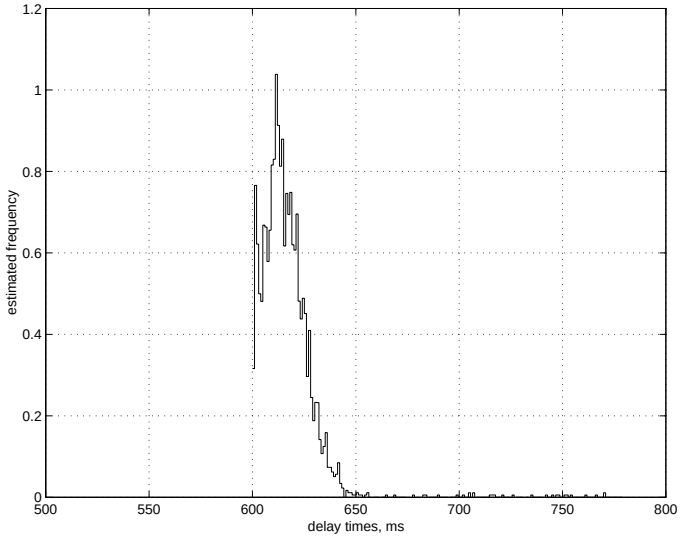

Fig. 8. Estimated density of Y

suppression. Additionally as far as we know, no-one has used real measurement data to enhance and verify their models.

Some early analytical work on the buffer size requirements for packetised voice is summarised by Gopal *et al.* [GWM84]. One often cited piece of work is Barberis [Bar81]. As part of this work he assumes the delays experienced by packets of the same talkspurt are i.i.d according to an exponential distribution $p(t) = \lambda e^{-\lambda T}$ where $1/\lambda$ is the average transmission delay and standard deviation. M.K. Mehmet Ali *et al.* in their work on buffer requirements [AW86] model the arrival process as a Bernoulli trial with probability $[1 - F(j, n - j + 1)]$ of the event "no arrival yet" at each interval up to its arrival. The outcome of the trial is represented by the random variable $k(j, n)$:

$$k(j,n) = \begin{cases} 1 \text{ if packet j has arrived at or before time n} \\ 0 \text{ otherwise.} \end{cases}$$

Ferrandiz and Lazar in [FL88] look at the analysis of a real time packet session over a single channel node and compute its performance parameters as a function of their model primitives. They do not use any Markovian assumptions, rather an approach which uses a series of overload and under-load periods. During overload packets are discarded. They derive an admission control scheme based on an average of the packet arrival rate. Van Der Wal *et al.* derive a model for the end to end delay for voice packets in large scale IP networks [vdWMK99]. Their model includes different factors contributing to the delay but not the arrival process of audio packets per se. The mathematical model described here is also discussed in the book [Kaj02].

7 Conclusions

We have addressed the problem of modelling the arrival process of a single packet audio stream. The model can be used to produce packet audio streams with characteristics, at least, quite similar to the particular measurements we have obtained. The model is suitable for generating streams both with and without silence suppression applied at the source, and in addition, the case where packets are lost.

The work can be generally applied to research where modelling arriving packet audio streams needs to be performed. A natural next step is to use the arrival model presented here for an evaluation of the jitter buffer performance, such as investigating waiting times and possible packet loss in the jitter buffer. We observed from our model that the interarrival times are negatively correlated. This will have an impact on the dynamics and performance of a jitter buffer. With an accurate model, based on real data measurements, a realistic traffic generator could be constructed. In separate work we have gathered nearly 25,000 VoIP measurements from ten globally dispersed sites which we could utilise for 'parameterising' a model, depending on the desired scenario.

References

[AW86] Mehmet M. K. Ali and C. M. Woodside. Analysis of re-assembly buffer requirements in a packet voice network. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 233–238, Bal Harbour (Miami), Florida, April 1986.

[Bar81] Giulio Barberis. Buffer sizing of a packet-voice receiver. *IEEE Transactions on Communications*, COM-29(2):152–156, February 1981.

[FL88] Josep M. Ferrandiz and Aurel A. Lazar. Modeling and admission control of real time packet traffic. Technical Report Technical Report Number 119-88-47, Center for Telecommunications Research, Columbia University, New York 10027, 1988.

[GWM84] Prabandham M. Gopal, J. W. Wong, and J. C. Majithia. Analysis of playout strategies for voice transmission using packet switching techniques. *Performance Evaluation*, 4(1):11–18, February 1984.

[Kaj02] Ingemar Kaj. *Stochastic Modeling for Broadband Communications Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[vdWMK99] Walm van der Wal, Mandjes Michel, and Rob Kooij. End-to-end delay models for interactive services on a large-scale IP network. In *IFIP*, June 1999.

Paper C

Olof Hagsand, Ian Marsh and Kjell Hanson.
Sicsophone: A Low-delay Internet Telephony Tool. In *29th Euromicro Conference*, Belek, Turkey, September 2003.

"cause we were never being boring,
We had too much time to find for ourselves,
And we were never being boring,
We dressed up and fought, then thought: make amends,
And we were never holding back or worried that,
Time would come to an end"

Pet Shop Boys - Being boring

Sicsophone: A low-delay Internet telephony tool

Olof Hagsand[1], Ian Marsh[2] and Kjell Hanson[3]

[1] LCN Laboratory, IMIT, Royal Institute of Technology, Sweden
olofh@kth.se
[2] SICS AB, Stockholm, Sweden
ianm@sics.se
[3] Prosilient Software AB, Stockholm, Sweden
kjell@prosilient.com

Abstract. The end to end delay is a critical factor in the perceived quality of service for Voice over IP applications. Sicsophone is a complete VoIP system that couples the low level features of audio hardware with a standard jitter buffer playout algorithm. Using the sound card directly eliminates intermediate buffering as well as providing fine control over timers needed by a soft real-time application such as VoIP. A statistical based approach for inserting packets into audio buffers is used in conjunction with a scheme for inhibiting unnecessary fluctuations in our system. We also present mouth-to-ear delay measurements for selected VoIP applications and show that several hundreds of milliseconds can be saved by using the techniques described in this paper. A prototype for both UNIX and Windows platforms has been implemented, demonstrating that our system adapts to network conditions whilst maintaining low delays.

1 Introduction

Users of interactive VoIP applications demand low latency conversations. Replaying packetised audio requires that sufficient packets are available to the application in order to avoid gaps or glitches. The digital to analog conversion of sampled voice requires strict, synchronous timing despite the fact that the network and operating system may disrupt the process. The most common method to solve this problem is to introduce a small intermediary buffer between the decoded audio stream and the audio hardware which allows packets to be available for playout. Of course withholding packets instead of immediately playing them increases the total delay for a VoIP user. However, the longer packets can be delayed, the more resilient the receiver is to adverse network conditions. We should already point out that Sicsophone is a working implementation and that the algorithmic complexity is an important factor. We motivate this approach with real delay experiments and results. Hence the goal is not to compare the merits of various playout algorithms, this has been covered by many researchers, rather to give some insight into which issues are important when realising these schemes in real systems.

In this paper we refer to the mouth-to-ear delay as the total one way delay experienced by two speakers including the analog-digital-analog conversion. By jitter we mean the variability in the packet delay. This variability is the reason we need to buffer packets, thus our work focuses on how to detect and compensate for packet jitter in an efficient manner. Our solution is to insert packets directly into the memory of the sound card relieving the need for time consuming data copying operations or context switching. This approach saves precious time, avoids scheduling problems but requires careful buffer management.

Figure 1 illustrates the complete path of audio samples from a microphone at a sender to the loudspeaker at a receiver. Traditionally, a sender writes voice samples to the operating system which are subsequently sent across the network to a receiving host. At the receiver, data is read from the operating system interface where it is the responsibility of the application to adjust the buffer size as required, the traditional data path is shown by the solid lines in the figure.

In our approach we use the buffering available on sound cards and copy the packet payloads directly into this area. Therefore, we save copying the data to and from the application, plus not performing the de-jittering in the application. We describe our approach in the context of DirectSound [BD98] subsystem on the Windows platforms. It is important to point out that our approach is not confined to this architecture, a ring buffer with pointer support is sufficient to realise the ideas presented in this paper (alternatives for UNIX include [Riz97] or [Ree98]). However we describe the system using DirectSound as it is known to many developers, and was used in our experimental evaluation. It is important to add that we assume the end system is not under heavy load or consider Sicsophone as a hard real-time system.

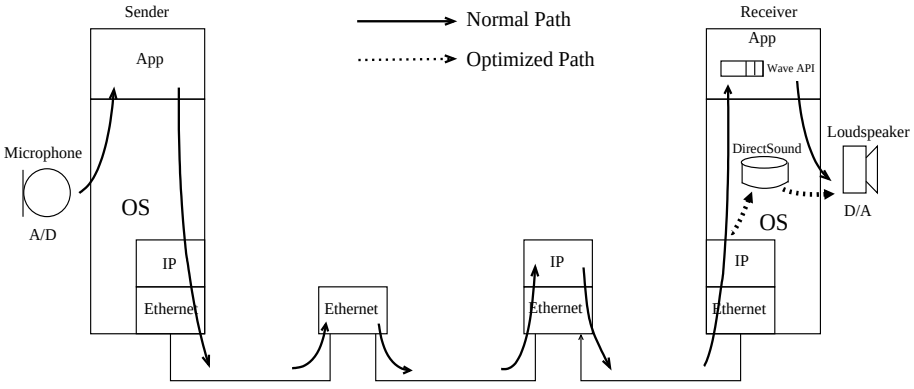

Fig. 1. Sicsophone audio delivery path

If we now look at the steps a receiver must take in order to replay packetised audio from a network in more detail, Table 1 shows four such typical steps. Firstly, de-packetisation, removes the IP and UDP headers and passes the data-

gram together with a Real Time Protocol (RTP) payload to a VoIP application. This step takes a few milliseconds on most systems. This step involves copying the data from the operating system to the application. Step two is to decode the sound samples, this is dependent on the compression scheme as well as the packet size used. Typically this takes from a few milliseconds to tens of milliseconds. Step three is the absorption of network delays through buffering. This is typically under the control of the Internet telephony application. Step four is delivery to the sound application, which usually means copying the data back to the operating system for it to copy the data (again) to the sound system. Our goal was to consolidate some of these steps into a single step, saving the time of intermediary buffering and context switching. We refer to this approach, solely for definition by its software name, Sicsophone.

Step	Process	Overhead	Depends On
1	De-packetisation	10 - 50	Pact. Size
2	Decoding	10 - 50	Coding
3	Buffer Delay	5 - 200	Network
4	Delivery	5 - 120	End System

Table 1. Typical receiver incurred delays (ms)

The remainder of this paper is organised in the following manner; Section 2 forms the main body of this work, the low-level adaption of playout buffers using ring buffers. Section 3 presents results of Sicsophone's performance for mouth-to-ear tests. We also give comparisons of the playout delay of Sicsophone against idealised cases. Section 4 is a description of related efforts with which this paper has commonalities, and we round off the work with some conclusions in Section 5.

2 End-system adaption to jitter

2.1 Buffering issues

In this section we outline some issues associated with the data buffering scheme we have chosen. Our goal is to save time by avoiding data copying, setting up direct memory access (DMA) transfers ahead of time, using simple data structures and inserting de-jittered audio packets directly into the memory of the sound card. Direct memory access is used to move data from memory to the sound card and vice versa without intervention of the CPU. Using DMA does not directly provide a time saving, as it can take some time to set up the transfer. However once it is done, the transfer can be done much quicker and more efficiently. This offers significant time savings over posting an interrupt for every packet, particularly in the older (and non-DirectX) versions of the Windows operating systems.

One potential problem of using the sound card memory as a buffer is it could be overrun by packets arriving too quickly. Modern sound cards however are equipped with megabytes of RAM to store down-loadable sound samples, DirectSound can allocate buffers up to this physical size when a hardware buffer is initialised. We do however keep the buffer from being overrun by mechanisms explained in Section 2.3. Another potential cause of overrun or underrun is misaligned or drifting clocks, there is no mechanism in Sicsophone to combat this.

Another important but often overlooked issue is mixing. For an application like VoIP where the voice channel is stopped and started continuously, we would like to minimise this setup time. Valuable time can be lost by setting up mixers for software and hardware buffers where we normally do not want to mix audio from different sources. Therefore we allocate a DirectSound primary buffer to give better delay characteristics, as it does not need to be mixed before output to D/A conversion.

The coding scheme used is another issue. Packets have to be decoded before insertion into the buffer if they are not in PCM format. However using PCM allows us to DMA the payload into the sound card memory without any audio format conversion. This is the *fastest* path from packet reception to playout. It is however possible to support other audio formats, however they require extra CPU cycles for decoding the audio, and a small buffer to hold the data before and after decoding. Using buffers in this manner makes the assumption a certain number of bytes in the buffer corresponds to a well defined playout time. This is the case for coding such as PCM but not for highly compressed audio formats.

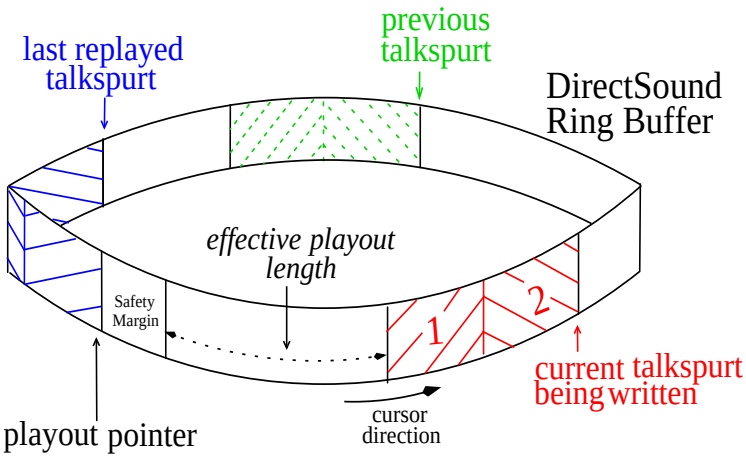

Fig. 2. DirectSound buffer structure

Figure 2 shows the interface offered by DirectSound. Data is written at the write pointer and replayed by the trailing playout pointer. The read and write pointers are updated by the system, and continuously encircle the buffer. Read-

ing and writing the pointers requires that the system almost instantaneously updates their current positions. Some of the older operating systems used in our tests did not give sufficiently fine granularity over the positions of the timers. In Sicsophone they are used as **both** timers and pointers. They function as a timer by indicating if a packet is too late. If the read pointer has already passed the point where a packet should be, and it has not been written, then we know that the next packet is late. Insertion is simply a modulo operation and a buffer copy. In order not to replay old data written into the buffer when no packets are being sent, we must write 'silence' samples into the ring. This is because some background noise is needed so that the aware the communication path is still open. Pure silence can be quite disconcerting.

Used as pointers, the read and write pointers give memory locations where data is read from or written to depending on the operation to be performed. Given these pointers it is simple to adjust the effective buffer length. It is the position of the read pointer relative to the write pointer. The closer the read pointer is to the write pointer the shorter the effective buffer length will be. Note there is a small margin of 15ms in front of the read pointer to allow data that has been written to be "ready" for playback. Use of this safety margin is recommended by Microsoft.

To give a concrete example, using an estimate of the network delay and its variation, described previously, we insert packets at a specified "distance" ahead of the read pointer. Therefore a translation from milliseconds to bytes is needed; $bytes = (samples/sec \cdot bits/sample \cdot P_i)$ / 8000. For example, if one substitutes 8000 for samples/sec, 8 bits per sample and 200ms for the playout point this equals 1600 bytes. This means that the write pointer can simply be set 1600 bytes in front of the read pointer. The safety margin should also be added to this value, which corresponds to an additional 125 bytes. To re-iterate once the playout point has been calculated, it is trivial to insert packets into the buffer, no complex data operations are needed.

2.2 Fast startup adaption

In an adaptive VoIP application we normally consider changing the buffer size during a silence period so as not to introduce audible glitches in the analog audio stream. Since the goal of this work is to produce a low-delay VoIP tool we would like to keep the buffer length as small as possible. However in the startup phase we have little knowledge of the network condition and therefore have to use default values for the network delay[1]. We therefore adjust the buffer length after monitoring only a few packets to find a fast estimate quickly.

Figures 3 and 4 show packet delay in the jitter buffer during the start up phase of Sicsophone as an example. The y-axis shows the waiting time in the buffer (in ms) and the x-axis shows the number of packets received, sorted by the time spent in the buffer, note this is **not** the sequence number. It only indicates the number of packets and their respective waiting times.

[1] Sicsophone's defaults are initial values of a 20ms minimum and an initial maximum of 60ms.

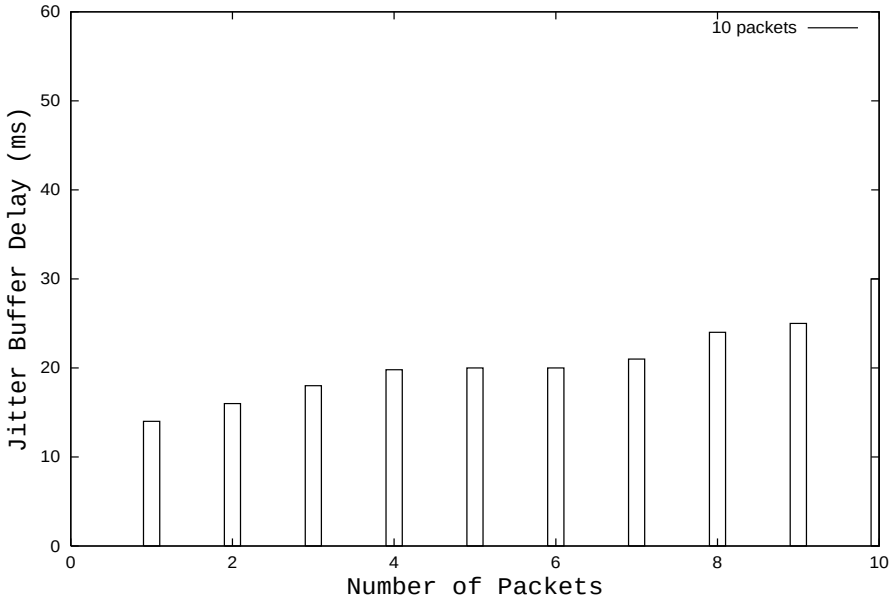

Fig. 3. Jitter for the first ten packets

Figure 3 shows the buffer state after ten packets have been received and Figure 4 after an additional 40 packets have arrived, the original ten are shown with somewhat bolder lines. After ten packets were stored, the time spent in the buffer varied between 14 and 30 ms whereas after 50 packets have been received the median delay incurred is approximately 20 ms.

Fast adaption is worthwhile during the start up phase of a VoIP session. The alternative approach is to be conservative in the start up phase and have long playout buffers until a value for the playout point can be calculated or a RTCP report received. In the presence of delay spikes [RKTS94] we can re-estimate the jitter value quickly. Since the goal of this work is to produce a low delay VoIP tool we chose quick adaption. Furthermore, usually there are sufficient silence periods during the startup phase of a conversation to perform fast adaption. In the case where there are none (such as call waiting, i.e. music playing) we adjust the buffer when a packet is excessively delayed or if there is a loss. Failing these possibilities we adjust the buffer length and possibly induce an audio glitch.

2.3 Bounding the estimated network delay

Sudden increases in the network delay can cause problems for a VoIP application. They typically result in a sequence of dropped packets whilst the receiver buffer is estimated and resized. A spike is referred to a sudden and rapid increase in the network delay which is typically short lived, often less than one round trip time. One solution is to track the increase in the delay and adjust the buffer

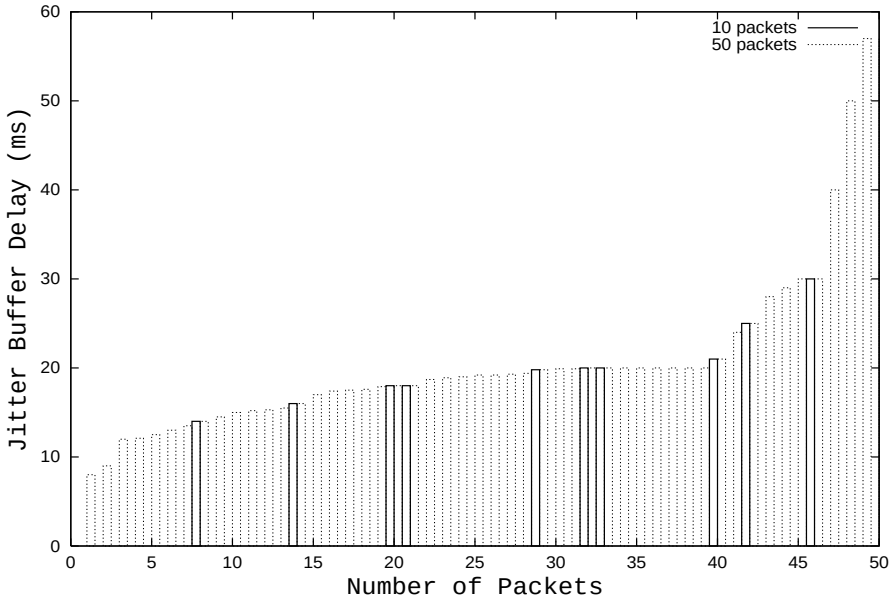

Fig. 4. Arrival of 10 and 50 packets

length accordingly. The alternative is to include the values of the jitter estimate but not to adapt the buffer size directly. It is because of this temporal property that has led us to be more conservative to network conditions after setup and not to adapt the DirectSound buffer length to sudden and transient changes in the network delay.

We have implemented a system which bounds the delay jitter estimate. As stated, the spikes are not completely ignored but we do not react immediately to their presence. The estimated jitter value should vary between an upper $Qmax_i$ and a lower $Qmin_i$ (see Figure 5) bound in a range, where $Qmin_i < d_i < Qmax_i$. If the running estimate breaks either of the boundaries we re-calculate the new buffer length, taking into account the value of the spike, but reset the mean estimate to the middle value of this new range. The values of $Qmin$ and $Qmax$ are calculated using a simple running average method. Figure 5 shows an example of a receiver jitter buffer during a call between two machines on a local network. The y-axis shows the jitter buffer length and the x-axis the sequence number. The system starts with Q_{min} and Q_{max} set to the default values of 20ms and 60ms for the minimum and maximum bounds respectively. At the bottom of the figure we show the range breaks as stars to highlight them. We have found this scheme to work relatively well, in the given trace there were only 14 breaches of the range from over 3600 packets (less than 0.5%). This example shows only a LAN example, however WAN facets are included in the next section. More importantly we did not make unnecessary changes to the DirectSound primary buffer.

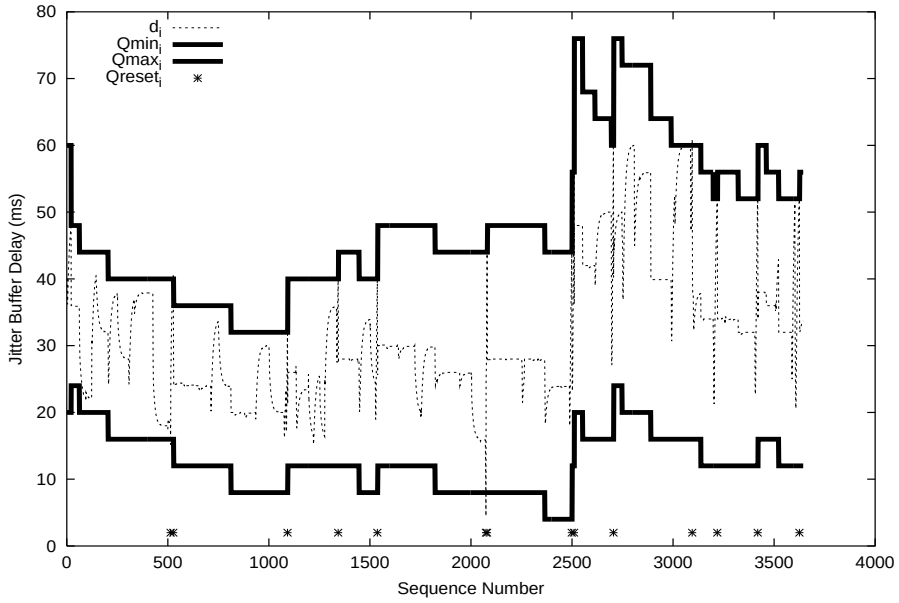

Fig. 5. Bounded jitter buffer playout delays

3 Evaluation and results

We divide this part into two sections, the first gives the total delay of popular VoIP tools compared to Sicsophone in a laboratory environment, with basically no, or little, network delay. Secondly, we show the performance of the playout algorithm using data taken from Internet measurements. This allows us to account for Sicsophone's WAN delay by comparing it with the best possible playout delay. This is done by post-processing measurement data. We chose to give the results in this manner to estimate the total mouth-to-ear delay by including both the WAN and LAN delays. In other words, the first set incorporates the delay due to coping with the operating system and the second with the network conditions.

3.1 Mouth-to-ear measurements

The delay reduction by Sicsophone is the main result of this paper. We performed one way mouth-to-ear measurements with a range of VoIP tools with the results summarised in Table 2. It's important to state that no parameter tweaking of these tools was done, we used their default installation values. The experimental setup used was as shown in Figure 1. We used a signal generator which generated a 1Hz square signal. The square wave serves as a trigger, the signal is packetised and sent over the IP network and played back through the loudspeaker at the

destination. The square wave is detected by an oscilloscope and the difference in time between the waves was measured.

Audio Tool	Latency (ms)
Sicsophone prototype	25-100
Ericsson Lanphone	300
Vocal Internet Phone 4.5 (SB)	450-550
Vocal Internet Phone 4.5 (PJ)	580-620
NetMeeting 2.1 (SB)	620
NetMeeting 2.1 (PJ)	750
VAT 3.4 (Solaris)	1200
RAT 3 (Solaris)	1500

Table 2. Mouth-to-ear latency measurements (SB=SoundBlaster and PJ=PhoneJack)

The measurements were done using a signal generator feeding a sender and an oscilloscope to measure the time difference between the sender and receiver. In retrospect it would better to add measurements that stressed the playout buffer, by using real speech rather than just a pulse. This was chosen so as to simply trigger (and calculate) the end-to-end delay. We can see that there are large variations between the various applications. One important result of this paper is to highlight the design of end systems for VoIP applications. Considerable time savings, 10's to 100's of milliseconds, can be saved by using an approach similar to the one described.

3.2 Comparison with ideal playout conditions

In the introduction we have eluded to a jitter buffer playout algorithm essentially has to tradeoff either delay or loss depending on the arrival process. Low delay implies a short playout buffer, incurring higher packet loss due to late arrivals. Whilst longer buffers reduce loss, they introduce delay into the system. When comparing the performance of algorithms it makes sense, therefore, to consider *both* loss and delay.

Figures 6 and 7 show the results for two Internet trace files. To calculate the optimal playout point we order all the packets and remove the 1% with the highest delay. We then calculate the delay needed to play the remaining 99% of the packets resulting in the delay for 1% packet loss, this process is repeated up to 25% packet loss (although in practice more than 10% would be deemed unacceptable for PCM). Figures 6 and 7 show Sicsophone's playout delay performance in comparison with an optimum.

In Figure 6 Sicsophone is about 50ms from the ideal playout point and remains more or less constant as the packet loss increases. For a given loss rate, e.g. 5%, Pinto and Christensen [PC99] quote a slightly lower delay than Sicsophone, 72ms compared to 98ms. The result in this case is due to the large variation

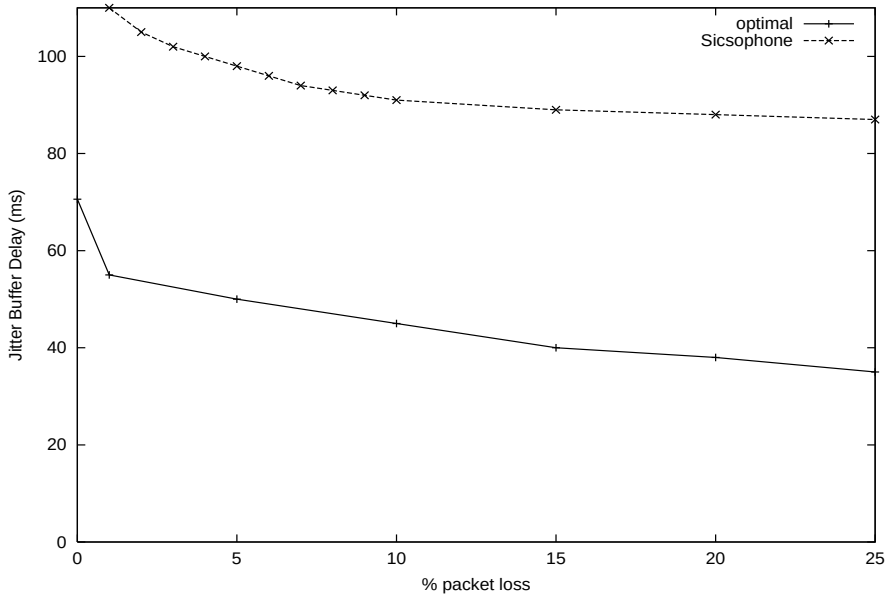

Fig. 6. Playout delays for a trace from UCI, California to INRIA, France

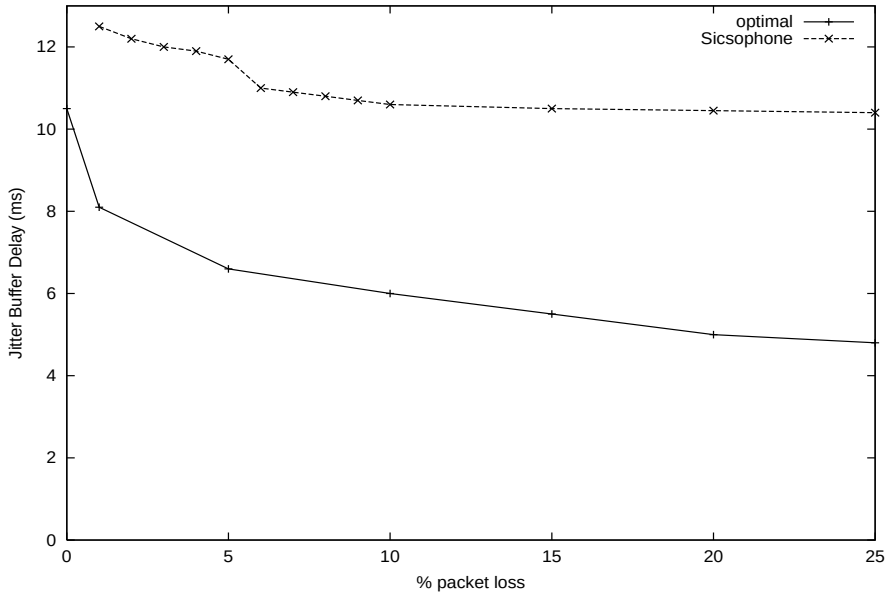

Fig. 7. Playout delays for a trace from Amherst, Mass to GMD, Berlin

of jitter (± 20ms), which makes it hard to settle to a constant value for the minimum buffer length, which can be verified by looking at the absolute jitter. A second test is shown in Figure 7 which shows a trace from the University of Massachusetts in Amherst to GMD in Berlin. In this case the jitter is lower and the difference between the optimal and Sicsophone is less than 5ms. We have shown worst and best cases from the measurement data available at that time.

4 Related work

The early 90's produced a surge in packet audio playout research. One of the first efforts to implement a voice application on an IP network with an adaptive buffer playout strategy was NeVoT [Sch92]. The playout algorithm implemented in Sicsophone is almost identical to NeVoT [Sch92]. They used a variation estimate similar to the one given earlier, however they make a slight distinction for the first packet in a talkspurt and subsequent ones. The playout for the first packet is delayed longer due to lack of information on the network state after the silence period. Our work shares theirs in the choice of a ring buffer for buffering packets, only we perform the copying by using DMA transfers directly rather than copying the data from the application to the operating system. VAT (Visual Audio Tool) [JM92] was a well known VoIP tool that implements a playout buffer similar to the one described, including a circular buffer to hold the packets before playout. We use an additional scheme to prevent the jitter estimates from varying too rapidly plus focus on the efficient insertion of packets into the playout buffer. Moon *et al.* [RKTS94] present four different playout algorithms for packet audio. They calculate an estimate of the network delay and jitter as an average from all the packets received. The authors highlight the jitter spikes we mentioned and also do not adapt the buffer size to these spikes. Pinto and Christensen [PC99] describe an algorithm for jitter compensation based on the target packet loss rate. Their "gap based" approach compares the current playout time with the arrival time and calculate a gap for both early and late packets. They compare the current playout delay, for any particular talkspurt in progress, with an optimal playout delay. This optimal theoretical delay is defined as minimum amount of delay to be added to the creation time of each packet which would result in a playout of a talkspurt at the given loss rate. Our calculation of the optimal playout is similar to the one described in this paper. Luigi Rizzo describes a generic sound card driver for FreeBSD [Riz97]. Aspects of this work resembles ours, in particular, handling of timers, DMA transfer and buffer size allocation. They include hooks to use the driver for VoIP applications, one such example is a select() call which can be scheduled to return only when a certain amount of data is ready for consumption. Rosenberg et. al in [RQS00] looked at combining target-based playout algorithms in conjunction with FEC schemes, and propose a number of new playout algorithms based on this coupling. Kouvelas and Hardman in [KH97] keep the flow of audio constant during high operating system load by using buffering in the audio hardware. They also look at reducing the amount of buffering in the application by keeping the buffers in the application

as small as possible. In our case we try and totally eliminate it dramatically by only using the sound card's storage possibilities.

5 Conclusions

In this paper we have shown how careful buffer management combined with a simple statistical playout scheme can reduce mouth-to-ear delay for VoIP applications. As stated at the start of this paper, the delay is one of the most important factors in the perceived QoS and hence has been the focus of this work. The results are encouraging as the mouth-to-ear delay of Sicsophone on a LAN is around 50ms on a Windows NT system with DirectX 8.0. We also include an estimate of the delay induced by network conditions using a standard playout algorithm. We have proposed a system which tries to reduce the perceived mouth-to-ear delay of real-time packet audio communication.

References

[BD98] Bradley Bargen and Peter Donnelly. *Inside DirectX*. Microsoft Press, 1998.

[JM92] Van Jacobson and Steve McCanne. VAT - LBNL Audio Conferencing Tool, July 1992. Available at http://www-nrg.ee.lbl.gov/vat/.

[KH97] Isidor Kouvelas and Vicky Hardman. Overcoming workstation scheduling problems in a real-time audio tool. In *Proc. of Usenix Winter Conference*, Anaheim, California, January 1997.

[PC99] J Pinto and K Christensen. An algorithm for playout of packet voice based on adaptive adjustment of talkspurt silence periods. In *Proceedings of the IEEE 24th Conference on Local Computer Networks*, pages 224–231. ACM, October 1999.

[Ree98] Dicken Reed. A new audio device driver abstraction. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 1998.

[Riz97] L. Rizzo. The FreeBSD audio driver. *Lecture Notes in Computer Science*, 1356, 1997.

[RKTS94] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 680–688, Toronto, Canada, June 1994. IEEE Computer Society Press, Los Alamitos, California.

[RQS00] Jonathan Rosenberg, Lili Qiu, and Henning Schulzrinne. Integrating packet FEC into adaptive voice playout buffer algorithms on the internet. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Tel Aviv, Israel, March 2000.

[Sch92] Henning Schulzrinne. Voice communication across the Internet: A network voice terminal. Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.

Paper D

Olof Hagsand, Kjell Hanson and Ian Marsh.
Measuring Internet Telephony Quality: Where are we today? In *Proceedings of IEEE Globecom: Global Internet*, pages 1838-1842, Rio De Janeiro, Brazil, December 1999.

"Heaven holds a sense of wonder,
and I wanted to believe"

Delerium featuring Sarah McLachlan - Silence

Measuring Internet telephony quality: Where are we today?

Olof Hagsand[1], Ian Marsh[2] and Kjell Hanson[3]

[1] LCN Laboratory, IMIT, Royal Institute of Technology, Sweden
olofh@kth.se
[2] SICS AB, Stockholm, Sweden
ianm@sics.se
[3] Prosilient Software AB, Stockholm, Sweden
kjell@prosilient.com

Abstract. Users of Internet telephony applications demand good quality audio playback. This quality is largely dependent on the instantaneous network conditions as well as the time of day. In this paper we describe a scheme for measuring network connections as well as a motivation for including a new metric when judging quality. Our tests included a wide range of geographically distributed sites. Our results give useful feedback to users and operators of IP Telephony networks and important information for developers of Voice over IP applications.

1 Introduction

The Internet makes no *guarantee* about the delivery of data in real time to applications such as Internet telephony. Due to the shared nature of many of the resources as well as propagation delays this is a difficult, if not impossible, task. All is not lost however, under good conditions in a well dimensioned network timely delivery of packets is achievable.

Due to the number of simultaneous connections and relative times across connected sites, the instantaneous quality of connections can vary dramatically. The goal of this paper is to show the benefit of taking and measuring quality so action can be taken immediately (if possible) or in the future. Also we wanted to produce a report on "how well are we doing today".

Internet telephony and voice over IP applications are already being used on the Internet. Most transport data using the Real Time Protocol (RTP)[SCFJ96] report statistics using RTCP, the Real Time Control Protocol. One function of RTCP is to report statistics back to the sender on the received quality of its receiver(s). In combination, the sender and receiver can obtain information on the following: the packet loss since the last message, the total packet loss in this session, the variance in packet arrival (jitter) and an estimate of the round trip time.

Many studies and measurements have been conducted on the Internet. Two of the most recent cited (and complete) works were done by Vern Paxson in [Pax96,Pax97]. Work dedicated to the measurement of audio data included Jean

Bolot et. al in 1995 in [BCG95] where the authors performed loss measurements and developed a model indicating that forward error correction (FEC) would rectify most loss situations. In 1997 work done by Mexemchuck and Lo in [ML97] defines the quality of a connection is as the fraction of the time that a channel is free of distortion for intervals that are long enough to transmit active speech segments.

In this work we look at both loss and delay and give measurements including these metrics, additionally we look closer at the interarrival variance or *jitter*, in particular the statistical distributions for a number of Internet sites. The results not only impact on quality but also include useful hints for application writers when designing adaptive playout schemes.

2 Measurements

2.1 Motivation

In addition to the simple quality argument, we have give a number of motivating reasons for performing wide area measurements:

- To perform an up-to-date series of measurements on today's Internet as far as voice is concerned.
- To dismiss hard limits for delay regarding Internet telephony quality [Int93]. We agree that delay should be minimized as much as possible without sacrificing playout quality, however it should not be a *definition* of the quality.
- The number of users and their access to the Internet, particularly through wireless connections, will affect the traffic considerably, we wanted to make measurements before this next quantum jump occurs.
- On well provisioned links we can carry packet audio without additional QoS mechanisms such as Integrated or Differentiated services. Finding the limit of what a provisioned link is, is one goal of this work.
- Clearly the time which one sends audio packets clearly influences the quality one receives. Traffic varies according to a daily rhythm. The respective times of both ends of the communication should be taken into account when presenting the measurement results.
- Work done in [Pax97] cites asymmetric routes. Therefore propagation delays, packet losses and possibly quality might be affected. This is expanded further in Section 2.4.

2.2 Delay

As most telephony users can be sensitive to high delays, we include a delay estimation. Measuring the real end-to-end delay is non trivial. Problems with synchronizing clocks and the best we can achieve is an estimation of the delay. We can assume the end-to-end delay is equal to the sum of some small packetization delay at the sender, the propagation delay and the receiver induced delay. Artificially introduced delay at the receiver is necessary to achieve smooth playout of

packets due to interarrival differences contributed by the statistical multiplexing of packet switched networks. This is a well known artifact and is explained in more detail in [RKTS94] [MKT95] bounds and [Sch92]

The packetization delay on most operating systems is relatively small approximately 20ms according to [SSO83]. The propagation delay we can approximate from RTT timer in RTCP and dividing by two. A point to note is if the route is asymmetric then this may be an approximation, however other methods exist to calculate the one-way delay. The delay added by buffering can be estimated by counting the number of bytes that the packet must wait in a buffer [Jac94].

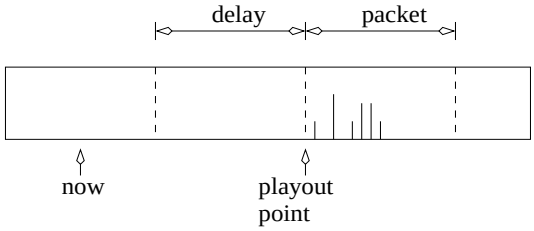

Fig. 1. Delaying packets for playout

The time the packet must wait (in milliseconds) is dependent on the sample rate and the number of bits per sample:

$$\text{delay} = \frac{sampling \;\; rate \;\cdot\; bytes}{samples \;\; per \;\; sec \;\cdot\; bits \;\; per \;\; sample}$$

for a typical 160 byte payload at 8000 Hz this becomes:

$$\text{delay} = \frac{8000 \cdot 160}{8000 \cdot 8} = 20\, ms$$

2.3 Interarrival times

Interarrival times can be obtained by finding the difference of consecutive arrivals. The variance of interarrival times is found by continually updating the differences from previous packets. The mean variation is usually calculated for each packet and this value is modified by a smoothing constant normally less than 1. The smoothing constant can be tuned to give more or less weight to recent arrivals. This mean variance or *jitter* gives a good indication of the connection conditions. Plots of some interrarival times for connections are shown in Figure 5. The jitter value shown in these figures is sent by the receiver to the sender as part of the RTCP receiver report.

2.4 Network asymmetry

Asymmetry is a problem in today's Internet. Paxson reported that in 1995, half of the 40,000 measurements he took had at least one different city in each direction

of a bidirectional flow. There are implications for IP telephony, particularly loss, where one receiver may report much poorer quality than the other. We show one example of an asymmetric plot of the interarrival times for a trace in Figure 9.

2.5 Software tool and measurements

We have implemented an IP telephony tool capable of sending RTP and RTCP packets. We can record and play PCM encoded files. The textual log files are in the following format:

```
I 919339994 729295
T 919339995 732069
E 919339996 38883 172 32869 0 160
E 919339996 68883 172 32869 1 320
--------------------------------
1 2         3     4   5     6 7
(Header: V=2 P=0 X=0 CC=0 M=0 PT=101)
```

Fig. 2. Log file format

3 Measurement setup

Our selected test sites consisted of one national (Luleå, Northern Sweden), one continental (Cambridge, UK), one trans-Atlantic site in the US, (Amherst, Cambridge) and finally one site in Buenos Aires in Argentina. For all measurements, a recorded session was sent from the remote site to a local site in Sweden. In the bidirectional tests the executions where performed simultaneously. Table 3 gives the details of the recorded session.

Trace File	Value
File Size	584480 bytes
Duration	70 secs
Payload	160 bytes (20 ms)
With Silence Suppression.	3643 packets
W/o Silence Suppression	2064 packets
% Transmitted	56.6 %

Fig. 3. Trace file properties

4 Results and Discussion

Figure 4 shows the delay as reported by RTCP at the receiver for hosts in Argentina and the US. It is quite evident that the connection to the US is much

better. Table 6 shows a summary of the delay for the four sites. Figure 5 shows

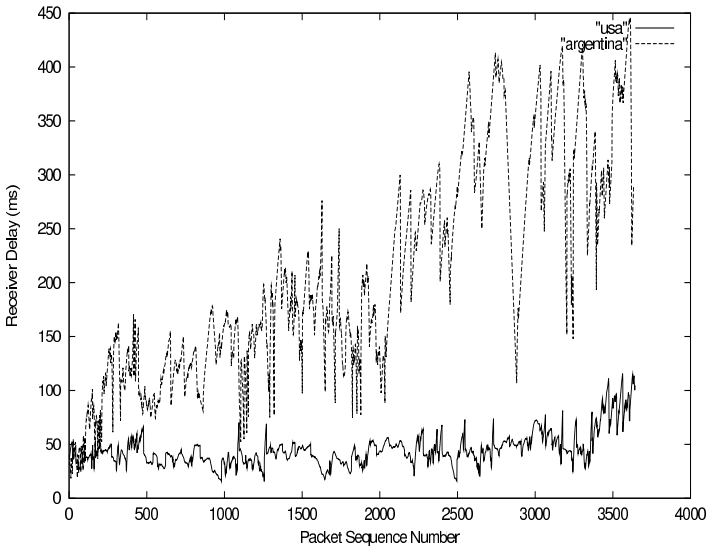

Fig. 4. Receiver Delay for 3 Remote Sites

the variance in the packet interarrivals. Packets from the national site (Luleå) form a peak which would result in less jitter and consequently less frequent changes in the playout buffer size. The converse is true for the connection to Argentina, where the variance typically ranges from 20ms to 150ms.

Figure 6 shows an estimate of the propagation delay (based on the RTCP reporting) and an estimation of the delay incurred by the end system. Within RTCP it is possible to extract the time a report was held at a particular node. Figure 7 shows the number of hops and the measured jitter for the same four sites. Finally figures 8 and 9 show losses in both directions for a conversation from Stockholm to Buenos Aires, in the first figure it is clear to see the higher loss rate from Argentina to Sweden. Similarly so for the interarrival times, which show much more variance in the South America to Europe direction.

5 Conclusions/Future Work

In this short paper we have presented some criteria for evaluating the quality of Internet Telephony connections. On average, with the exception of the South American site, reasonable quality Internet telephony can be supported. Obviously more connections need to be tested, but can be done with the tools we

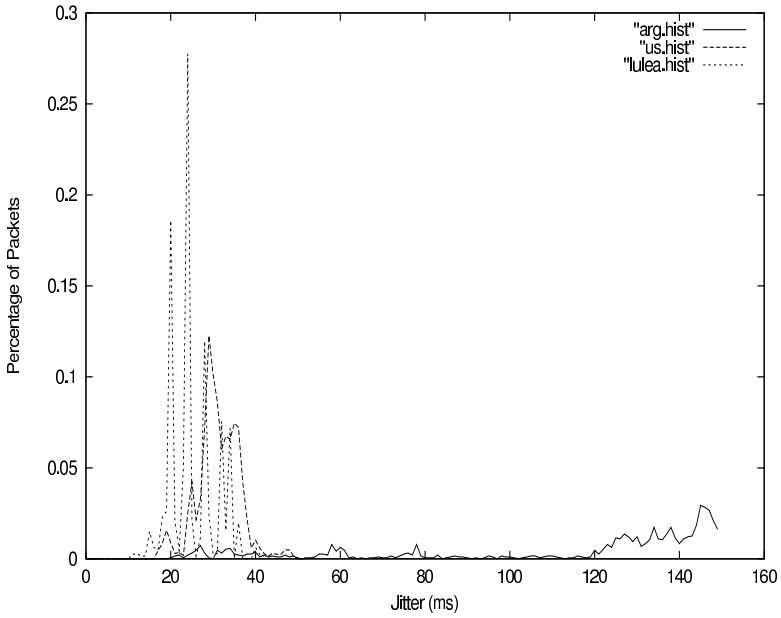

Fig. 5. The interarrival histogram for two sites

Source	Propagation	Receiver	Total
Luleå (Sweden)	12.13	42.14	54.27 ms
Cambridge (UK)	33.3	54.39	87.68 ms
Amherst (US)	57.3	50.82	108.12 ms
Buenos Aires (Arg)	273.0	119.12	392.12 ms

Fig. 6. Mean Delay Values

Source	Hops	Jitter
Luleå (Sweden)	9	5.12 ms
Cambridge (UK)	15	7.38 ms
Amherst (US)	15	11.25 ms
Buenos Aires (Arg)	18	117.69 ms

Fig. 7. Mean Jitter Values

have started to develop. One proviso, the sites used were located on academic networks, further tests would have to be done on commercial networks.

We have developed a tool that probes, gathers and produces statistics based on the RTP and RTCP protocols. Additionally we have included time-driven

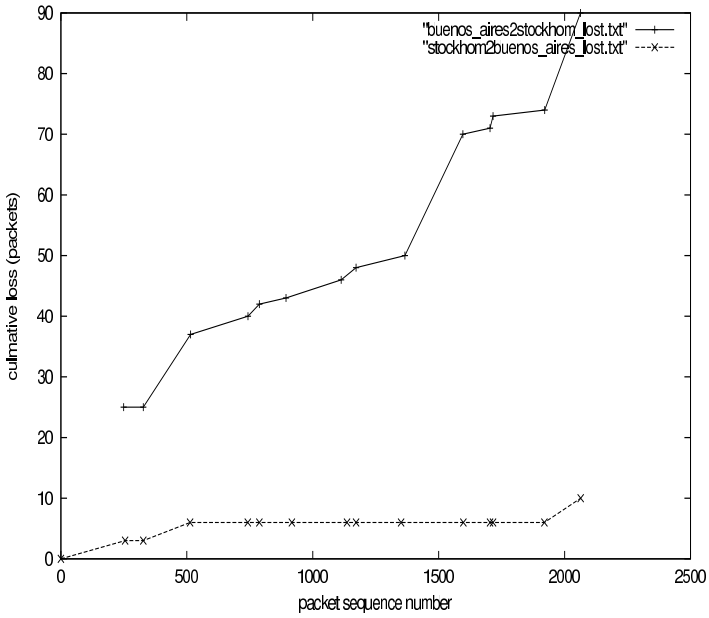

Fig. 8. Bidirectional Loss (Cumulative)

tracing as well as looked at the bidirectional (and hence asymmetric) quality. We plan to extend the tool to operate as a daemon in which the tests can run without intervention. This also allows any site to connect to any other therefore producing a full mesh of connections rather than our current centralized system.

We would like to thank Thiemo Voigt and Bengt Ahlgren for their comments on this paper, Ericsson and Telia for their financial support. Also thanks to Steve Pink, Jim Kurose and Pablo Giambiagi for the accounts that we have used.

References

[BCG95] J. Bolot, H. Crepin, and A. Garcia. Analysis of audio packet loss in the internet. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Lecture Notes in Computer Science, pages 163–174, Durham, New Hampshire, April 1995.

[Int93] International Telecommunication Union (ITU). Transmission systems and media, general recommendation on the transmission quality for an entire international telephone connection; one-way transmission time. Recommendation G.114, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, March 1993.

[Jac94] Van Jacobson. Multimedia conferencing on the Internet. In *SIGCOMM Symposium on Communications Architectures and Protocols*, London, England, August 1994. Tutorial slides.

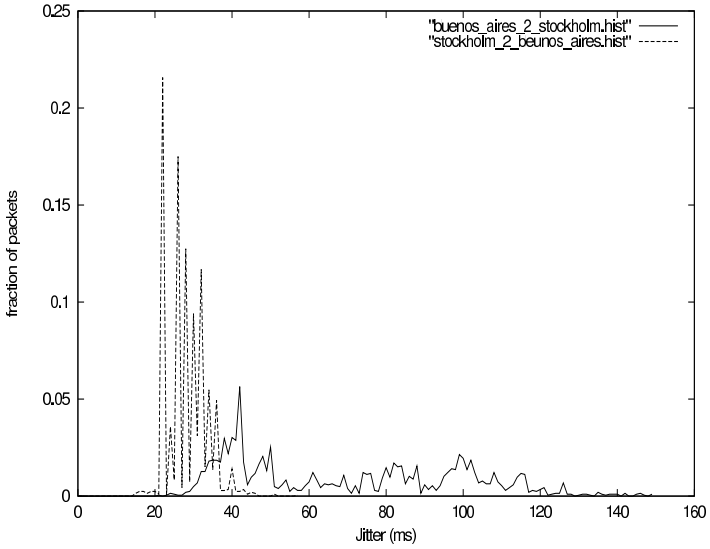

Fig. 9. Histogram of the bidirectional interarrival times

[MKT95] Sue B. Moon, Jim Kurose, and Don Towsley. Packet audio playout delay adjustment algorithms: performance bounds and algorithms. Research report, Department of Computer Science, University of Massachusetts at Amherst, Amherst, Massachusetts, August 1995.

[ML97] N. F. Maxemchuk and S. Lo. Measurement and interpretation of voice traffic on the internet. In *Conference Record of the International Conference on Communications (ICC)*, Montreal, Canada, June 1997.

[Pax96] Vern Paxson. End-to-end routing behavior in the internet. In *SIGCOMM Symposium on Communications Architectures and Protocols*, Stanford, California, August 1996.

[Pax97] Vern Paxson. End-to-end internet packet dynamics. In *SIGCOMM Symposium on Communications Architectures and Protocols*, Cannes, France, September 1997.

[RKTS94] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 680–688, Toronto, Canada, June 1994. IEEE Computer Society Press, Los Alamitos, California.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications. Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, January 1996.

[Sch92] Henning Schulzrinne. Voice communication across the Internet: A network voice terminal. Technical Report TR 92-50, Dept. of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.

[SSO83] Daniel C. Swinehart, L. C. Stewart, and S. M. Ornstein. Adding voice to an office computer network. In *Proceedings of the IEEE Conference on*

Global Communications (GLOBECOM), pages 392–398 (11.4), San Diego, California, November 1983. IEEE. also Xerox Report CSL-83-8.

Paper E

Ian Marsh and Fengyi Li.
Wide Area Measurements of VoIP Quality. In *Quality of Future Internet Services 2003*, October, 2003, Stockholm, Sweden.

"I feel extraordinary,
something's got a hold on me,
I get this feeling I'm in motion,
A sudden sense of liberty"

New Order - True Faith

Wide Area Measurements of Voice Over IP Quality

Ian Marsh[1], Fengyi Li[2] and Gunnar Karlsson[2]

[1] SICS AB, Kista S-164 29, Sweden
ianm@sics.se
[2] LCN Laboratory, IMIT, Royal Institute of Technology, Sweden
d97-fli@nada.kth.se, gk@imit.kth.se

Abstract. Time, day, location and instantaneous network conditions largely dictate the quality of Voice over IP calls. In this paper we present the results of over 18000 VoIP measurements, taken from nine sites connected in a full-mesh configuration. We sample the quality of the routes on a hourly basis by transmitting a pre-recorded call between a given pair of sites. We repeat the procedure for all nine sites during the one hour interval. Based on the obtained jitter, delay and loss values as defined in RFC 1889 (RTP) we conclude that the VoIP quality is acceptable for all but one of the nine sites involved. We also conclude that VoIP quality has improved marginally since we last conducted a similar study in 1998.

1 Introduction

It is well known that the users of real-time voice services are sensitive and susceptible to variable audio quality. If the quality deteriorates below an acceptable level, or is too variable, users often abandon their calls and retry later. Since the Internet is increasingly being used to carry real-time voice traffic, the quality provided has become, and will remain an important issue. The aim of this work is therefore to disclose the current quality of voice communication at selected end-points on the Internet.

It is intended that the results of this work will be useful to many different communities involved with real-time voice communication. We now list some groups to whom this work might have relevance. First end users can determine which destinations are likely to yield sufficient quality. When deemed insufficient they can take preventative measures such as adding robustness, for example in the form of forward error correction to the outgoing packets. Operators can use findings such as these to motivate upgrading links, adding control admission schemes, rerouting traffic or even implementing QoS mechanisms where poor quality is being reported. Network regulators can use these kind of measurements to verify the quality level that has been agreed upon or even to settle disputes. Speech coder designers can utilise the data as input for a new class of codecs, of particular interest are designs which yield good quality in the case of bursty packet loss. Finally, researchers could use the data for their own investigations.

The structure of this paper is as follows: Section 2 begins with some background on the quality measures we have used in this work namely, loss, delay and jitter. Following on from the quality measures, section 3 gives a description of the methodology used to measure the parameters. In section 4 the results are presented (and condensed) into a single table. In section 5 the related work is given, comparing results obtained in this study with other researchers' work. This is considered important as it indicates whether quality has improved or deteriorated since those studies. Section 6 rounds off with some conclusions and a pointer to the data we have collated.

2 What Do We Mean by Voice over IP Quality?

Ultimately, users judge the quality of voice transmissions. Organisations such as ETSI, ITU, TIA, RCR plus many others have detailed methods to assess voice quality. Assigning a quality measure involves replaying coded voice to both experienced and novice listeners and asking them to adjudge the perceived quality. Measuring the quality of voice data that has been transmitted across a wide area network is more difficult primarily due to the delay. The network inflicts its own impairment on the quality of the voice stream. By measuring the delay, jitter and loss of the incoming data stream at the receiver, we can provide some indication on how suitable the *network* is for real-time voice communication.

It is important to point out we did not include the quality contribution of the end systems in this study. This is because the hardware was different at each site, even with our own software Sicsophone [HMH03]. But also, to avoid including the delays of our own application into the measurements. In order to assess the delay contribution of each end system it would be difficult without isolation tests. We chose to use simple A-law PCM coding to maintain a theoretically known coding/decoding delay.

The quality of VoIP sessions can be quantified by the network delay, packet loss and packet jitter. We emphasise that these three quantities are the major contributors to the perceived quality as far as the *network* is concerned. The G.114 ITU standard states that the end-to-end one way delay should not exceed 150ms [RG98]. Delays over this value adversely effect the quality of the conversation. An alternative study by Cole and Rosenbluth state that users perceive a linear degradation in the quality up to 177ms [CR02]. Above this figure the degradation is also linear although markedly worse. As far as the packet loss is concerned, using speech coding schemes such as A-law or μ-law coding, tests have shown that the mean packet loss should not exceed 10% before an appreciable loss in quality. Note that a loss rate such as this does not say anything about the distribution of the losses. As far as the authors are aware of, no results exist that state how jitter solely can affect the quality of voice communication. Work on jitter and quality are often combined with loss or delay factors. When de-jittering mechanisms are employed, the network *jitter* is manifested into application *delay or loss*. The application must hold back a sufficient number of packets in order to ensure smooth, uninterrupted playback of speech. To sum-

marise, we refer to the quality as a combination of delay, jitter and loss. It is important to mention we explicitly do not state how these values should be combined. The ITU E-model is one approach but others exist, therefore we refer the interested reader to the references such as [LE01] and [KKI91].

3 Simulating and Measuring Voice over IP Sessions

Our method to measure VoIP quality is to send pre-recorded calls between globally distributed sites. Through the modification of our own VoIP tool, Sicsophone, the intervening network paths are probed by a 70 second pre-recorded 'test signal'. The goal of this work is therefore to report in what state the signal emerges after traversing the network paths.

Nine sites have been carefully chosen with different hop counts, geographic distances and time-zones to obtain a diverse selection of distributed sites. One important limitation of the available sites was they were all located at academic institutions, which are typically associated with well provisioned networks. Their locations are shown in Figure 1. The sites were connected as a full mesh allowing

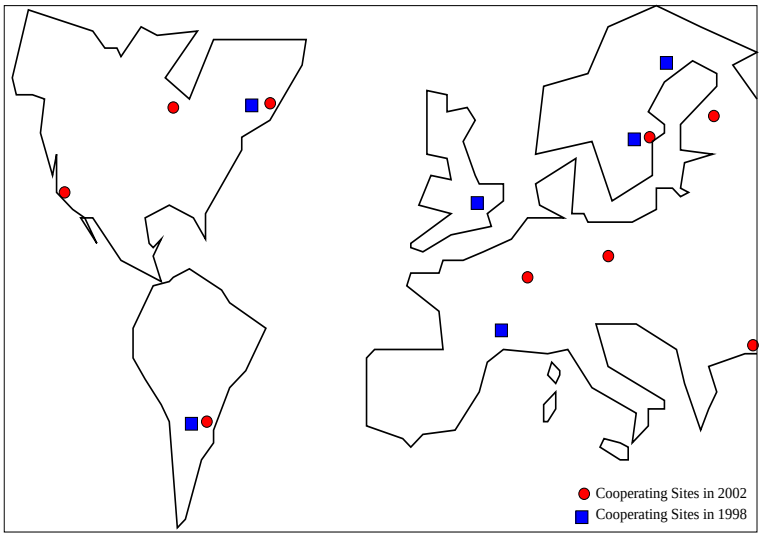

Fig. 1. The nine sites used in the 2002 measurements are shown with circles. The six depicted with squares show those that were available to us in 1998, three remained unchanged during the past four years.

us, in theory, to measure the quality of 72 different Internet paths. In practice, some of the combinations were not usable to port blocking there were four such cases. Bi-directional sessions were scheduled on a hourly basis between any two given sites. Calls were only transferred once per hour due to load considerations on remote machines.

In Table 1 we list the characteristics of the call we used to probe the Internet paths As stated, the call is essentially a fixed length PCM coded file which can be sent. The file consisted of a PCM coded recording of a read passage of text. Over a 15 week period we gathered just over 18,000 recorded sessions. The number of sessions between the nine sites is not evenly distributed due to outages at some sites, however we attempted to ensure an even number of measurements per site.

Test "signal"	
Call duration	70 seconds
Payload size	160 bytes
Packetisation time (ms)	20ms
Data rate	64kbits/sec
With silence suppression	2043 packets
Without silence suppression	3653 packets
Coding	8 bit PCM
Recorded call size	584480 bytes
Obtained data	
Number of hosts used (2003)	9
Number of traces obtained	18054
Number of data packets	32,771,021
Total data size (compressed)	411 Megabytes
Measurement duration	15 weeks

Table 1. The upper half of the table gives details of the call used to measure the quality of links between the sites. The lower half provides information about the data which was gathered.

3.1 A Networking Definition of Delay

We refer to the delay as the *one way network* delay. One way delay is important in voice communication, particularly if it is not the same in each direction. Measuring the one way delay of network connections without synchronised clocks however, is a non-trivial task. Hence many methods rely on round-trip measurements and halve the values, hence estimating the one way delay. We measured the network delay using the RTCP protocol which is part of the RTP standard [SCFJ96]. A brief description follows. At given intervals the sender transmits a so called "report" containing the time the report was sent. On reception of this report the receiver records the current time. Therefore two times are recorded within the report. When returning the report to the sender, the receiver subtracts the time it initially put in the report, therefore accounting for the time it held the report. Using this information the sender can calculate the round-trip delay and importantly, discount the time spent processing the reports at the receiver. This can be done in both directions to see if any significant anomalies exist. We quote the network delay in the results section as it explicitly does not

include any contribution from the end hosts. Therefore it is important to state the delay is not the mouth-to-ear delay, rather only the network delay.

3.2 Jitter - An IETF Definition

Jitter is the statistical variance of the packet interarrival time. The IETF in RFC 1889 defines the jitter to be the mean deviation (the smoothed absolute value) of the packet spacing difference between the sender and the receiver [SCFJ96]. Sicsophone sends packets of identical size at constant intervals which implies that $S_j - S_i$ (the sending times of two consecutive packets) is constant. The difference of the packet spacing, denoted D, is used to calculate the interarrival jitter. According to the RFC, the interarrival jitter should be calculated continuously as each packet i is received. The interarrival jitter J_i for packet i is calculated using the previous packet J_{i-1} thus:

$$J_i = J_{i-1} + (|D(i-1,i)| - J_{i-1})/16.$$

According to the RFC "the gain parameter 1/16 gives a good noise reduction ratio while maintaining a reasonable rate of convergence". As stated earlier, buffering due to jitter adds to the delay of the application. This delay is accounted for in the results we present. The real time needed for de-jittering depends on how the original time spacing of the packets should be restored. For example if a single packet buffer is employed it would result in an extra 20ms (the packetisation time) being added to the total delay. Note that packets arriving with a spacing greater than 20ms should be discarded by the application as being too late for replay. Multiples of 20ms can thus be allocated for every packet held before playout in this simple example. To summarise, the delay due to de-jittering the arriving stream is implementation dependent, thus we do not include it in our results.

3.3 Counting Ones Losses in the Network

We calculate the lost packets as is exactly defined in RFC 1889. It defines the number of lost packets as the expected number of packets subtracted by the number actually received. The loss is calculated using expected values so as to allow more significance for the number of packets received. For example 20 lost packets from 100 packets has a higher significance than 1 lost from 5. For simple measures the percentage of lost packets from the total number of packets expected is stated. As stated, the losses in this work *do not* include those incurred by late arrivals, as knowledge of the buffer playout algorithm is needed. Detailed analysis of the loss patterns is not given in the results section, we simply state the percentages of single, double and triplicate losses.

4 Results

The results of 15 weeks of measurements are condensed into Figure 2. The table

should be interpreted as an 11 by 11 matrix. The locations listed horizontally across the top of the table are the locations used as receivers. Listed vertically they are configured as senders. The values in the rightmost column and bottom row are the statistical means for all the connections **from** the host in the same row and **to** the host in the same column respectively. For example the last column of the first row (directly under "Mean") is the average delay to all destinations from Massachusetts (112.8ms).

Each cell includes the delay, jitter, loss, number of hops and the time difference prefixed by the letters D, J, L, H and T for each of the connections. The units for each quantity are the delay in milliseconds, the jitter in milliseconds, the loss as a percentage, the hops as reported by traceroute and time differences in hours. A '+' indicates that the local time from a site is ahead of the one in the corresponding cell and behind for a '-'. The values in parenthesis are the standard deviations. A NA signifies "Not Available" for this particular combination of hosts. The bottom rightmost cell contains the mean for all 18054 calls made, both to and from all the nine hosts involved.

The most general observation is the quality of the paths is generally good. The average delay is just below the ITU's G.114 recommendation for the end-to-end delay. Nevertheless at 136ms it does not leave much time for the end systems to encode, dejitter, decode and replay the voice stream. A one packet buffer would easily absorb the 4ms jitter. However the 150ms threshold can be seen as the lowest possible, some extra delay can be tolerated.

There are two clear groupings from these results, those within the EU and the US and those outside. The connections in Europe and the United States (and between them) are very good. The average delay between the US/EU hosts is 105ms, the jitter is 3.76ms and the loss 1.16%. Those outside fair less well. The Turkish site suffers from large delays, which is not surprising as the Turkish research network is connected via a satellite link to Belgium (using the Geant network). The jitter and loss figures however are low, 5.7ms and 4% respectively. The Argentinian site suffers from asymmetry problems. The quality when sending data to it is significantly worse than when receiving data from it. The delay is 1/3 higher, the jitter is more than twice it in the opposite direction and the loss is nearly four times higher than when sending to it. Unfortunately we could not perform a traceroute from the host in Buenos Aires, so we cannot say how the route contributed to these values.

We now turn our attention to results which are not related to any particular site. As far as loss is concerned the majority of losses are single losses. 78% of all the losses counted in all trace files were single losses whereas 13% were duplicate losses and 4% triplicate losses. For some connections (22 from 68), some form of packet loss concealment would be useful, as the loss is over 1% but nearly always under 10%.

Generally the jitter is low relative to the delay of the link, approximately 3-4%. This is not totally unexpected as the loss rates are also low. With the exception of the Argentinian site, the sites did not exhibit large differences in asymmetry and were normally within 5% of each other in both directions. It is

interesting to note that the number of hops could vary somewhat under the 15 week measurement period. Only very few ($< 0.001\%$) out of sequence packets were observed. Within [Li02] there are details of further tests, such as the effect of using silence suppression, using larger payload sizes and daytime effects. In summary no significant differences were observed in these tests. We can attribute this (and the good quality results) to generally well-provisioned academic networks.

5 Related Work

Similar but less extensive measurements were performed in 1998 [HHM99]. Only three of the hosts remain from four years ago, so comparisons can only be made for these routes (and towards Stockholm). An improvement, in the order of 5-10% has been observed for these routes. We should point out though, the number of sessions recorded four years ago numbered only tens per host, whereas on this occasion we performed hundreds of calls from each host. Bolot et. al. looked at consecutive loss for designing an FEC scheme [BCG95]. They concluded that the number of consecutive losses is quite low and stated that most losses are one to five losses at 8am and between one to ten at 4pm. This is in broad agreement with the findings in this work. Maxemchuk and Lo measured both loss and delay variation for intra-state connections within the USA and international links [ML97]. Their conclusion was the quality depends on the length of the connection and the time of day. We did not try different connection durations but saw much smaller variations (almost negligible) during a 24 hour cycle (see [Li02]). We attribute this to the small 64kbits per second VoIP session on well dimensioned academic networks. It is worthy to point out our loss rates were considerably less than Maxemchuks (3-4%). Dong Lin had similar conclusions [Lin99], stating that in fact even calls within the USA could suffer from large jitter delays. Her results on packet loss also agree with those in [BCG95], which is interesting, as the measurements were taken some four years later.

6 Conclusions

We have presented the results of 15 weeks of voice over IP measurements consisting of over 18000 recorded VoIP sessions. We conclude that the quality of VoIP is good, and in most cases is over the requirements of many speech quality recommendations. Recall that all of the sites were at academic institutions which is an important factor when interpreting these results as most universities have well provisioned links, especially to other academic sites. Therefore this work should not be over generalised. Nevertheless, the loss, delay and jitter values are very low and from previous measurements the quality trend is improving. We can only attribute this to more capacity and better managed networks than those four years ago. However some caution should be expressed as the sample period was only 15 weeks, the bandwidth of the flows is low compared to the available capacities and only sampled once per hour. We have a large number of sample sessions

so can be confident the findings are representative of the state of the network at this time. One conclusion is that VoIP is obviously dependent on the IP network infra-structure and not only on the geographic distance. This can be clearly seen in the differences between the Argentinian and Turkish hosts. Concerning the actual measurement methodology, we have found performing measurements on this scale is not an easy task. Different access mechanisms, firewalls, NATs and not having permissions on all machines, complicates the work in obtaining (and validating later) the measurements. Since it is not possible to envisage all the possible uses for this data we have made it available for further investigation.

Future work involves further improvements in collecting and analysing the data. During these measurements we did not save (but sent) the audio samples at the receiver, however future measurements will do so in order to capture the quality degradation to lost packets. Extending the measurement infra-structure to non-academic sites is also a natural progression of this work. Performing quality measures that include the end systems should also be considered, although how to include the heterogeneity of the end systems still remains unresolved.

References

[BCG95] J. Bolot, H. Crepin, and A. Garcia. Analysis of audio packet loss in the internet. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Lecture Notes in Computer Science, pages 163–174, Durham, New Hampshire, April 1995.

[CR02] R.G Cole and J.H Rosenbluth. Voice over IP Performance Monitoring. *ACM Computer Communication Review*, 2002.

[HHM99] Olof Hagsand, Kjell Hansson, and Ian Marsh. Measuring Internet Telephone Quality: Where are we today ? In *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, Rio, Brazil, November 1999.

[HMH03] Olof Hagsand, Ian Marsh, and Kjell Hanson. Sicsophone: A Low-delay Internet Telephony Tool. In *29th Euromicro Conference*, Belek, Turkey, September 2003.

[KKI91] Nobuhiko Kitawaki, Takaaki Kurita, and Kenzo Itoh. Effects of Delay on Speech Quality. *NTT Review*, 3(5):88–94, September 1991.

[LE01] B.M.Lines L.F.Sun, G.Wade and E.C.Ifeachor. Impact of Packet Loss Location on Perceived Speech Quality. In *Proceedings of 2nd IP-Telephony Workshop (IPTEL '01)*, pages 114–122, Columbia University, New York, April 2001.

[Li02] Fengyi Li. Measurements of Voice over IP Quality. Master's thesis, KTH, Royal Institute of Technology, Sweden, 2002.

[Lin99] Dong Lin. Real-time voice transmissions over the Internet. Master's thesis, Univ. of Illinois at Urbana-Champaign, 1999.

[ML97] N. F. Maxemchuk and S. Lo. Measurement and interpretation of voice traffic on the Internet. In *Conference Record of the International Conference on Communications (ICC)*, Montreal, Canada, June 1997.

[RG98] ITU-T Recommendation G.114. General Characteristics of International Telephone Connections and International Telephone Circuits: One-Way Transmission Time, Feb. 1998.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889, Internet Engineering Task Force, January 1996.
http://www.rfc-editor.org/rfc/rfc1889.txt.

receiver / sender	Massachusetts	Michigan	California	Belgium	Finland	Sweden	Germany	Turkey	Argentina	Mean
Massachusetts	*	D:38.0 (17.1) J:2.4 (1.7) L:0.1 (0.6) H:14 (+1) T:0	D:54.2 (15.8) J:2.4 (1.8) L:0.1 (0.9) H:19 T:-3	D:67.1 (15.5) J:3.6 (1.5) L:0.1 (0.8) H:11 T:+6	D:97.1 (2.6) J:2.5 (1.5) L:0.1 (0.8) H:15 T:+7	D:99.5 (8.5) J:3.2 (1.7) L:0.04 (0.2) H:21 T:+6	D:58.4 (5.0) J:4.5 (1.4) L:0.0 (0.0) H:17 (+3) T:+6	D:388.2 (43.2) J:10.4 (4.9) L:4.9 (4.7) H:20 T:+7	D:99.7 (4.9) J:19.9 (8.4) L:8.9 (7.2) H:25 T:+1	D:112.8 J:6.1 L:1.2 H:17
Michigan	*	D:36.4 (15.4) J:4.7 (0.8) L:0.0(0.2) H:14 (+1) T:0	D:40.4 (4.5) J:4.4 (0.8) L:0.2 (1.1) H:20 (+1) T:-3	D:63.5 (4.2) J:4.3 (0.7) L:0.0 (0.1) H:11 T:+6	D:88.2 (8.0) J:4.1 (0.7) L:0.1 (1.1) H:17 T:+7	D:86.7 (4.7) J:5.2 (0.6) L:0.1 (2.2) H:23 T:+6	D:63.6 (8.2) J:7.3 (1.9) L:0.2 (0.9) H:16 (+1) T:6	D:358.9 (44.9) J:5.6 (1.7) L:3.0 (1.9) H:20 T:7	D:112.1 (10.6) J:18.7 (7.9) L:6.5 (7.0) H:25 T:+1	D:106.2 J:6.8 L:1.3 H:18
California	*	D:54.5 (16.7) J:2.0 (1.0) L:0.1 (0.36) H:18 (+1) T:+3	D:40.6 (5.1) J:1.2 (0.6) L:0.1 (1.9) H:21 T:+3	*	D:106.0 (3.0) J:1.4 (0.8) L:0.6 (1.4) H:25 (+1) T:+10	D:108.0 (2.4) J:2.1 (0.9) L:0.2 (0.3) H:30 (+2) T:+9	D:81.5 (1.8) J:4.9 (1.5) L:2.8 (3.0) H:23 T:+9	D:386.9 (60.5) J:5.3 (1.7) L:4.4 (2.4) H:23 T:+10	D:123.9 (12.4) J:18.1 (9.9) L:8.9 (8.2) H:25 T:+4	D:122.2 J:4.6 L:2.2 H:23
Belgium	*	D:65.2 (10.1) J:1.6 (0.6) L:0.0 (0.0) H:16 T:-6	D:63.4 (3.3) J:0.6 (0.1) L:0.0 (0.0) H:17 T:-6	D:84.0 (1.3) J:0.9 (0.8) L:1.2 (1.0) H:23 T:-9	*	D:31.3 (0.6) J:0.9 (0.5) L:0.0 (0.0) H:17 T:+1	D:33.4 (0.2) J:1.6 (0.9) L:0.0(0.0) H:22 T:0	D:16.6 (10.4) J:3.4 (1.5) L:0.21 (0.7) H:13 T:0	D:341.1 (24.7) J:6.9 (2.0) L:3.8 (2.7) H:16 (+2) T:+1	D:136.5 (7.1) J:NA L:NA H:19 T:-5	D:96.4 J:2.0 L:0.6 H:17
Finland	*	D:97.8 (4.2) J:1.7 (0.8) L:0.0 (0.1) H:15 (+1) T:-7	D:86.8 (1.9) J:1.1 (0.6) L:0.0 (0.3) H:17 (+1) T:-7	D:109.9 (4.7) J:1.4 (0.8) L:0.7 (1.4) H:24 (+2) T:-10	D:30.7 (0.3) J:1.4 (0.6) L:0.1 (0.3) H:16 T:-1	*	D:13.6 (1.0) J:1.9 (0.9) L:0.0 (0.0) H:20 T:-1	D:26.8 (7.3) J:3.9 (1.1) L:0.0(0.0) H:20 (+1) T:-1	D:321.2 (39.3) J:3.4 (1.7) L:3.2 (1.7) H:17 (+2) T:0	D:161.5 (12.2) J:17.4 (8.2) L:7.5 (6.5) H:19 T:-6	D:106.3 J:4.1 L:1.4 H:18
Sweden	*	D:99.3 (8.8) J:3.0 (1.9) L:0.0 (0.0) H:22 (+1) T:-6	D:84.9(1.9) J:2.5 (2.0) L:0.03 (0.4) H:25 T:-6	D:105.6 (2.1) J:3.2 (1.96) L:0.1 (0.1) H:30 T:-9	D:33.3 (0.4) J:2.8 (1.6) L:0.1 (0.3) H:24 T:0	D:13.5 (0.5) J:2.4 (1.8) L:0.0 (0.01) H:21 T:+1	*	D:29.8 (12.8) J:4.8 (2.5) L:0.0 (0.0) H:25 T:0	D:322.2 (30.3) J:3.2 (1.49) L:2.9 (1.0) H:26 T:+1	D:165.6 (17.9) J:NA L:NA H:41 T:-5	D:107.8 J:2.8 L:0.4 H:26
Germany	*	D:63.5 (9.6) J:1.72 (0.7) L:0.0(0.0) H:15 T:-6	D:60.4 (0.5) J:0.7 (0.3) L:0.0 (0.0) H:16 T:-6	D:84.4 (1.0) J:1.8 (0.7) L:2.5 (1.9) H:22 T:-9	D:11.1 (0.2) J:0.8 (0.3) L:0.0 (0.0) H:12 T:0	D:27.8 (7.3) J:1.0 (0.5) L:0.0 (0.0) H:17 T:+1	D:29.2 (7.6) J:1.5 (0.6) L:0.0 (0.0) H:22 T:0	*	D:300.7 (39.7) J:4.8 (2.1) L:3.7 (2.5) H:16 T:+1	D:149.8 (15.6) J:NA L:NA H:18 T:-5	D:90.9 J:1.6 L:0.8 H:17
Turkey	*	D:379.1 (47.1) J:8.6 (0.7) L:8.1 (2.8) H:18 (+1) T:-7	D:387.9 (35.5) J:8.9 (1.2) L:8.0 (2.9) H:20 T:-7	D:410.9 (43.9) J:8.8 (2.5) L:7.6 (6.8) H:19 T:-10	D:330.2 (28.6) J:9.2 (2.0) L:7.10 (4.0) H:17 T:-1	D:318.9 (42.4) J:8.8 (0.6) L:7.8 (2.7) H:19 T:0	D:311.1 (8.3) J:9.1 (0.7) L:8.4 (3.1) H:25 T:-1	D:378.2 (49.3) J:10.7 (1.2) L:8.0 (3.1) H:16 T:-1	*	D:490.8 (26.0) J:NA L:NA H:18 T:-6	D:375.9 J:8.0 L:6.9 H:19
Argentina	*	D:117.0 (30.8) J:4.2 (2.0) L:0.5 (1.4) H:NA T:-1	D:146.7 (44.2) J:4.3 (2.3) L:0.5 (1.5) H:NA T:-1	D:152.0 (47.8) J:3.1 (2.4) L:0.6 (1.8) H:NA T:-4	D:NA J:4.2 (2.0) L:0.5 (1.4) H:NA T:+5	D:164.1 (27.2) J:3.9(2.2) L:0.0 (1.4) H:NA T:+6	D:160.9 (47.7) J:2.9 (0.8) L:0.0 (0.1) H:NA T:+5	D:180.5 (50.6) J:4.7 (1.5) L:0.1 (0.1) H:NA T:+5	D:NA J:6.0(1.2) L:5.8 (3.0) H:NA T:+6	*	D:115.2 J:4.2 L:1.1 H:NA
Mean	*	D:114.1 J:3.4 L:1.1 H:14	D:113.6 J:3.4 L:1.1 H:16	D:115.7 J:3.2 L:1.6 H:19	D:77.1 J:3.5 L:1.0 H:13	D:105.8 J:3.1 L:1.1 H:16	D:105.2 J:3.4 L:1.1 H:20	D:104.4 J:5.5 L:1.4 H:16	D:345.6 J:5.7 L:4.0 H:17	D:180.0 J:9.3 L:4.00 H:23	D:136.2 J:4.1 L:1.8 H:18

Fig. 2. A summary of 18000 VoIP sessions. The delay (D), jitter (J) and loss (L) for the nine sites. The delay and jitter are in milliseconds, the losses are in percentages. The number of hops (H) and time zones (T) in hours are also given. The means for each and all sites are given plus the standard deviations (in parentheses). An NA signifies 'Not Available'.

Paper F

Olof Hagsand, Ignacio Más, Ian Marsh and Gunnar Karlsson.
Self-admission control for IP telephony using early quality estimation. In *IFIP-TC6 4th Networking*, Athens, Greece, May 2004.

"All the things I haven't seen,
Once the final curtain has been raised,
The act we act is wearing thin,
The act we act is under my skin"

The act we act - Sugar

Self-Admission Control for IP Telephony using Early Quality Estimation

Olof Hagsand[1], Ignacio Más[1], Ian Marsh[2], and Gunnar Karlsson[1]

[1] LCN Laboratory, IMIT, Royal Institute of Technology, Sweden
olofh@kth.se, nacho@kth.se, gk@imit.kth.se
[2] Swedish Institute of Computer Science
ianm@sics.se

Abstract. If quality of service could be provided at the transport or the application layer, then it might be deployed simply by software upgrades, instead of requiring a complete upgrade of the network infrastructure. In this paper, we propose a self-admission control scheme that does not require any network support or external monitoring schemes. We apply the admission control scheme to IP telephony as it is an important application benefiting from admission control. We predict the quality of a call by observing the packet loss over a short initial period using an in-band probing mechanism. The quality prediction is then used by the application to continue or to abort the call. Using over 9500 global IP telephony measurements, we show that it is possible to accurately predict the quality of a call. Early rejection of sessions has the advantage of saving valuable network resources plus not disturbing the on-going calls.

1 Introduction

Quality of service in the Internet has been researched for the last twenty years, yet its introduction has been extremely slow. Differentiated services [BBC+98] was originally proposed in 1997 to overcome scalability problems of previous proposals. However, DiffServ is still not widely offered by Internet service providers, perhaps due to the required upgrade in the network infrastructure. Our proposal offers a light QoS for multimedia stream traffic, by a regulated admission of sessions, rather than a regulation of the flow rate per session. It can be argued it is more desirable to block a call that has little chance of being completed with adequate quality rather than allowing it to start and potentially degrade the system. Therefore, admitted sessions gain by having a high probability of being completed with acceptable quality. These properties can be successfully accomplished by using admission control.

The purpose of this paper is to devise an efficient and flexible admission control scheme for IP telephony. Although IP telephony is used as the example real-time application in this work, it should be clear that there are no inherent restrictions on the applicability of the admission control scheme. The admission control can be performed without explicit support from the network [SRC84]. The procedure is in–band probing [BKS+00], in which the first seconds of the

voice transmission are used as a probe stream. A new session is established only after estimating that the state of the network is acceptable. The receiver of the call measures the packet loss ratio of the first few seconds and estimates the packet loss probability. This estimated loss probability is compared to an acceptance threshold, which determines whether the session should be established or not. Loss levels above the threshold result in blocking the new session and the sender must wait before establishing a new session. Hence, ongoing calls are protected from new calls that could deteriorate the overall quality to an unacceptable level by placing additional load on the network. The admission control being proposed is related to the out-of-band probing scheme being developed in our group [Kar98,FKR00,MIK01,MFK01].

We claim that measurements can produce data useful for predicting future quality. However, it is important to state we use only packet loss as the quality indicator of a VoIP session in this work. Packets arrive at a receiver 50 times per second (assuming no loss) in our VoIP scheme [HMH03], so we have frequent sampling and network state observations. The measured loss after an initial number of seconds (zero to ten) is compared with the loss measured over the whole session. We use the correlation between the two measurements to determine how accurate the estimation is. This is possible as we have the whole session recorded at the receiver stored available for post processing.

The structure of this paper is as follows. The next section gives some background on how the empirical measurements were taken; we also describe how we measure the packet loss ratio for one call. Section 3 shows the results for all considered calls and offers a statistical analysis of the accuracy of the loss estimation for different initial time intervals, as well as blocking and error probabilities. Section 4 gives some conclusions of our work, some applications and pointers to future work. A preliminary version of this work was published in [BHK+03].

2 Measurement description

This paper uses the results of previous work where approximately 23000 VoIP calls were measured between hosts at nine academic sites [ML03]. The locations of the sites are shown in Figure 1. The sites were connected as a full-mesh, allowing us, in principle, to measure the quality of 72 different Internet paths. These paths represent large differences in timezones, hop counts and geographic distances.

The measurements were performed over a period of 15 weeks in the following way: A call between two hosts was initiated on an hourly basis between a sender and a receiver. The sender transmitted a sequence of pre-recorded speech samples at 64 kbps as a stream of RTP/UDP/IP datagrams. The receiver made a detailed log of the arrival process, recording the reception time of each datagram. The complete details of the measurements are described in [ML03].

2.1 Reducing the sample set

For the purposes of this paper, we needed a common basis for our analysis, and therefore selected a subset of the 23000 calls. We only used calls that experience loss, since loss free calls do not provide any extra information for our analysis: both the probing and the total loss rate are zero giving perfect correlation. A large percentage of the calls are in fact loss free which reduces the sample set somewhat. We attribute the large number of loss free sessions to the fact that the sites are located on well provisioned (academic) networks. This restriction resulted in a subset of 9683 calls. Despite this reduction, all nine sites are still represented in the subset.

2.2 Measuring a single call

Figure 2 shows the loss process of a sample call as observed by a receiver. The call was made between the Argentinian and Turkish sites. The figure shows a loss pattern that is representative of many other calls in the subset. The plot shows the number of lost packets on the y-axis versus time on the x-axis. It can be seen that the number of lost packets increases almost linearly as the call proceeds.

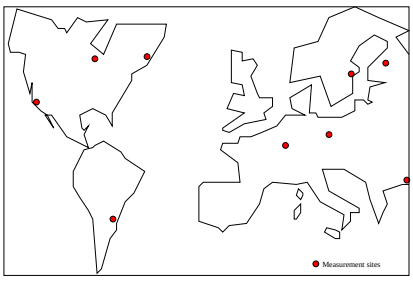

Fig. 1: Measurements were made between nine academic sites worldwide.

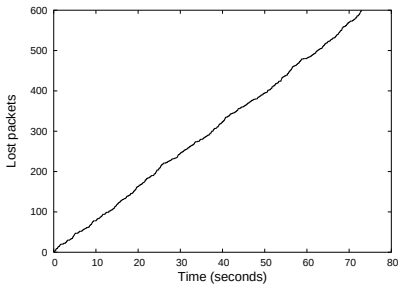

Fig. 2: Loss process of a single sample call between Turkey and Argentina.

Figure 3 shows the cumulative loss ratio for the same call. This ratio is defined as the number of lost packets divided by the number of sent packets. We show the cumulative plot to clarify how long we need to measure in order to obtain a good estimation of the final loss ratio. From the plot, we see that the final loss ratio for the complete call is approximately 18%.

In Figure 4 we show the first 20 seconds of the same call. From the figure, we see that the initial loss is approximately 14% after one second and 19% after ten seconds. These are early estimations of the final loss rate. We want to know how accurate such early estimations really are. Therefore we need to study the relation between the loss ratio of an initial part of the call and the loss ratio of the whole call duration.

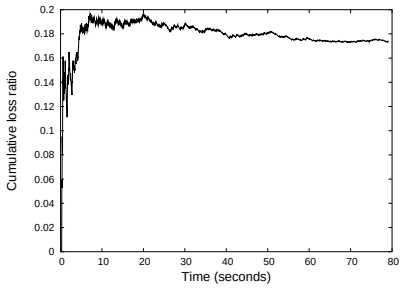

Fig. 3: Cumulative loss ratio of a single sample call between Turkey and Argentina.

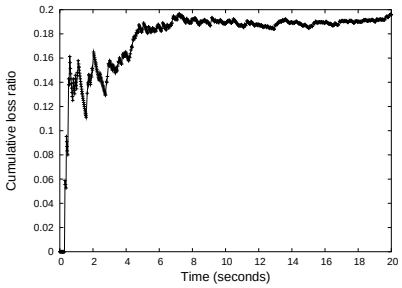

Fig. 4: Cumulative loss ratio of the same session showing only the initial portion of the call.

3 Analysis

In the preceding section, only one call was considered. In Figures 5 and 6, the loss ratio for the whole call is plotted versus the loss ratio of an initial interval for all calls in the selected subset. In the figures, every point represents one call. The plots show that as the initial interval increases, the points group closer around the line $y = x$. In other words, the correlation increases and the estimation improves.

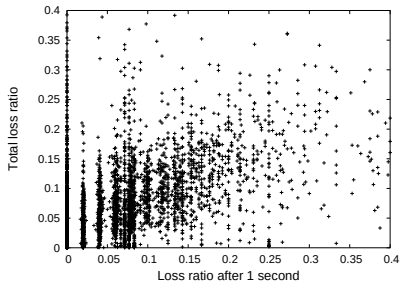

Fig. 5: Relation between the loss ratio after one second and the total loss ratio for all calls

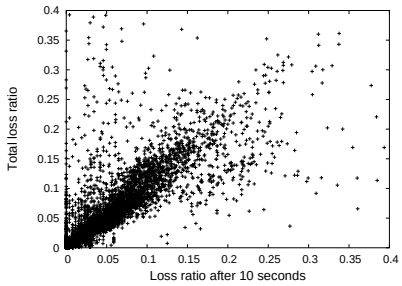

Fig. 6: Relation between the loss ratio after ten seconds and the total loss ratio for all calls

The plots in Figures 5 and 6 give an intuitive measure of the correlation between the loss ratio of the initial interval and the whole call. In order to evaluate more precisely the accuracy of the estimation, we computed the actual correlation coefficient as a function of the initial interval. The result is shown in Figure 7. It shows that the correlation coefficient increases as the probing interval increases. From the figure, we can clearly see that the correlation sta-

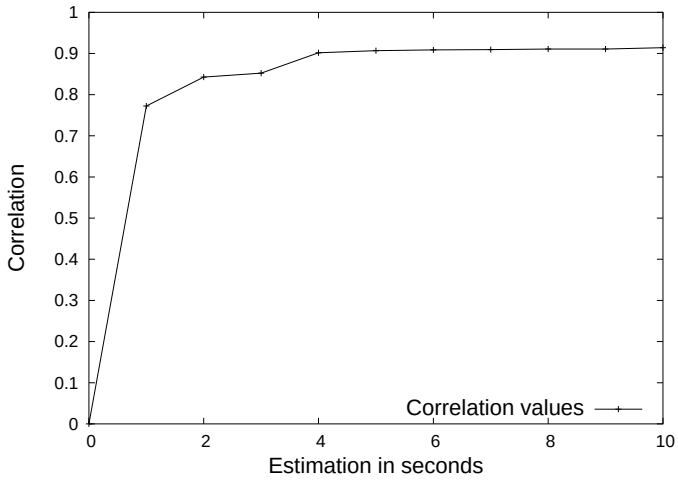

Fig. 7: The correlation coefficient as a function of the initial probing interval

bilizes after four seconds. This is important, because after this point no further estimates are necessary.

The relation between the loss ratio of an initial interval, l_p, and the loss ratio of the total call, l_t can be further examined by forming an error function, such as $l_t - l_p$, and analyzing it as a stochastic variable ϵ.

Figure 8 shows histograms of ϵ for probing intervals of one, four and ten seconds. In the histograms, positive values represent calls where the initial loss rate is smaller than the total loss rate, i.e., $l_p < l_t$. In other words, those calls experienced a higher packet loss after the probing, the quality of the calls deteriorated after the initial interval. Likewise, negative values represent calls where the initial loss rate is greater than the total loss rate, i.e., $l_p > l_t$. Note that the histograms represent probability density functions of ϵ that are not normally distributed.

Based on the values in the histograms we calculated the confidence intervals by counting the number of samples around $\epsilon = 0$ that sum up to the desired confidence level. The result is shown in Table 1. Based on the table, we can express to what degree we can trust an initial observation. For example, if we measure the loss ratio l_p of a call after four seconds, we can be 80% certain that the total loss of the call will be in the interval $[l_p - 1.8\%, l_p + 2.6\%]$.

While the confidence intervals may be useful in themselves to express confidence in an observed value, forming the cumulative distribution function (*cdf*) of ϵ is more useful when an upper bound on the final loss value is needed. This is typically the case in admission control scenarios, where we want to block calls that we believe will experience a loss higher than a certain threshold. Table 2 shows the *cdf* of ϵ. Using the table, we can make statements such as, given a probing loss and a confidence level, the final loss will be bounded by the probing

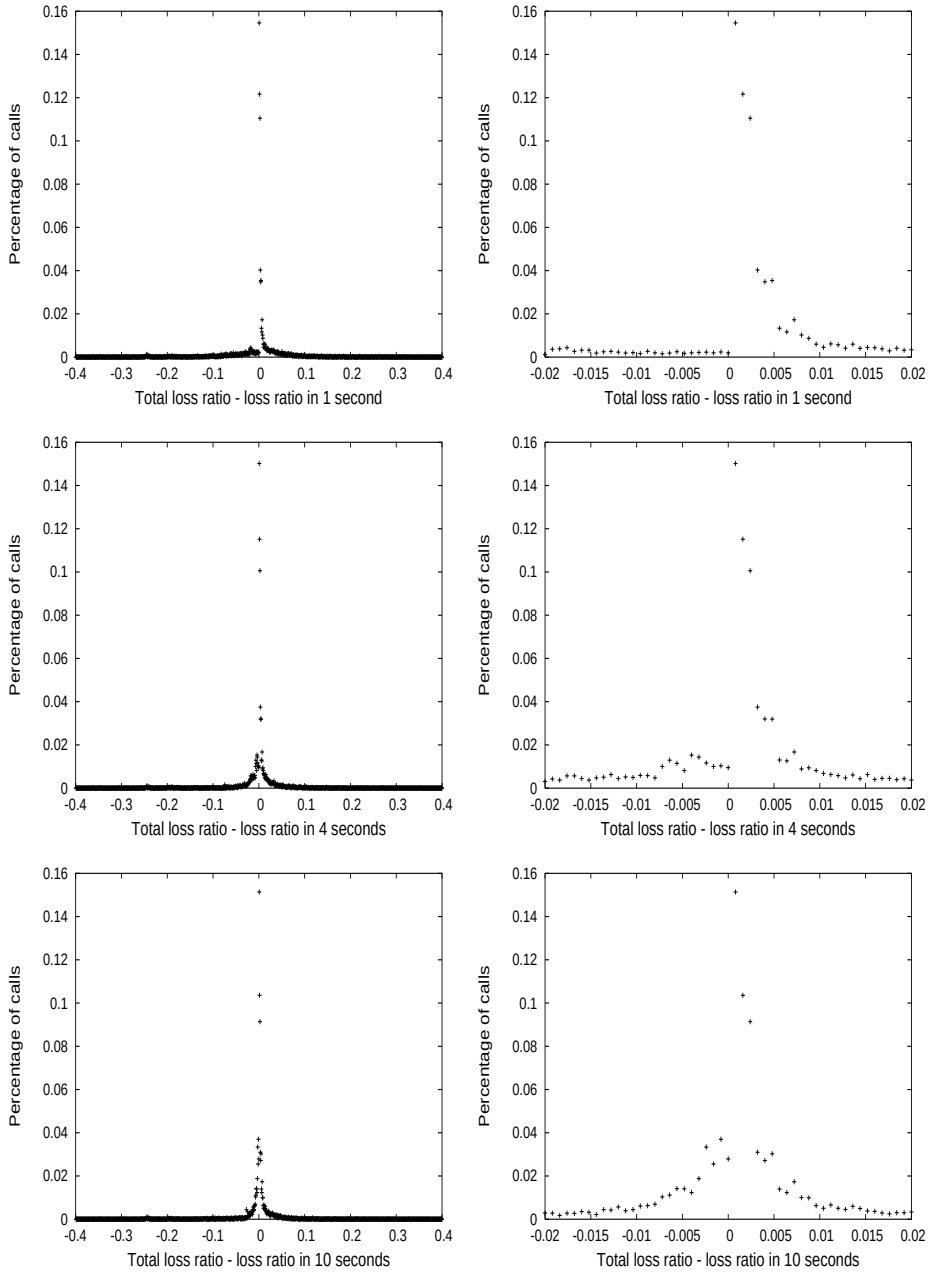

Fig. 8: Histograms of the error $\epsilon = l_t - l_p$ for initial probing intervals one, four and ten seconds. Each histogram is shown in full view on the left, while the right plot shows an enlarged region around $\epsilon = 0$.

Level	interval (1 second)	interval (4 seconds)	interval (10 seconds)
0.75	$[-0.0288, 0.0336]$	$[-0.0141, 0.0187]$	$[-0.0087, 0.0151]$
0.80	$[-0.0424, 0.0436]$	$[-0.0180, 0.0260]$	$[-0.0124, 0.0213]$
0.85	$[-0.0608, 0.0568]$	$[-0.0252, 0.0344]$	$[-0.0183, 0.0299]$
0.90	$[-0.0848, 0.0752]$	$[-0.0416, 0.0496]$	$[-0.0280, 0.0424]$
0.95	$[-0.1768, 0.1144]$	$[-0.1200, 0.0800]$	$[-0.0888, 0.0696]$
0.99	$[-0.4000, 0.2536]$	$[-0.3200, 0.2216]$	$[-0.2480, 0.2144]$

Table 1: Table showing confidence levels and intervals of the error function $\epsilon = l_t - l_p$ for probing intervals one, four and ten seconds.

loss plus a value given by Table 2. Figures 9, 10 and 11 show the *cdf* of ϵ in graphical form.

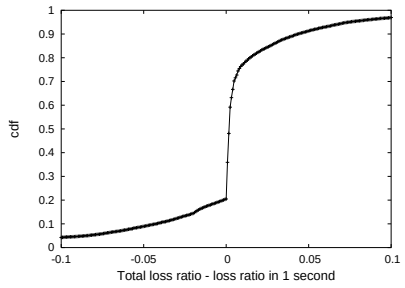

Fig. 9: Cumulative distribution function of $\epsilon = l_t - l_p$ for a probing interval of one second.

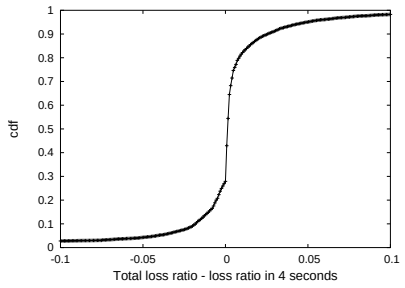

Fig. 10: Cumulative distribution function of $\epsilon = l_t - l_p$ for a probing interval of four seconds.

The *cdf* of ϵ can directly be used for admission control purposes. The table gives us the percentage of calls that have an error less or equal to the value of ϵ:

$$P(\epsilon < l_a - l_p) \geq confidence \quad level$$

For example, suppose the aim of a strict admission control scheme using four seconds probing is to drop calls that have a higher risk than 10% to surpass a pre-established loss rate l_a. Retrieving the value of ϵ from Table 2 shows that $l_a - 2.6\%$ is a good threshold. A more relaxed policy could have the aim to reject all calls that have more than 90% risk to surpass l_a. In that case, again using Table 2, the threshold is $l_a + 1.81\%$. The strict and relaxed policies outlined above both have drawbacks. With a strict policy, most bad calls ($l_t > l_a$) will be blocked, along with a large number of good calls ($l_t < l_a$). A relaxed policy admits most good calls, while admitting many bad calls.

Table 3 shows a classification of calls with respect to an admission control strategy: classes AG and AB represent calls that were admitted while classes

Confidence level	1 second	4 seconds	10 seconds
0.05	−0.0848	−0.0416	−0.0280
0.1	−0.0424	−0.0181	−0.0101
0.2	−0.0016	−0.0056	−0.0038
0.3	–	–	−0.0008
0.4	–	–	–
0.5	0.0018	0.0014	0.0009
0.6	0.0025	0.0020	0.0017
0.7	0.0042	0.0038	0.0030
0.8	0.0144	0.0086	0.0069
0.9	0.044	0.0260	0.0212
0.95	0.0752	0.0496	0.0424

Table 2: Table showing cumulative values of the error function $\epsilon = l_t - l_p$ for probing intervals one, four and ten seconds.

RG and RB represent calls that were blocked. Furthermore, classes AG and RB represent categories where the admission control decision was correct. Classes AB and RG represent decisions that were wrong. An admission control policy based on probing, needs to consider the tradeoff between classes.

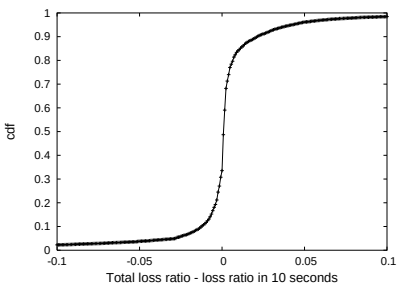

Fig. 11: Cumulative distribution function of $\epsilon = l_t - l_p$ for a probing interval of ten seconds.

	Good calls $l_t < l_a$	Bad calls $l_t > l_a$
Admitted $l_p < l_\alpha$	AG	AB
Rejected $l_p > l_\alpha$	RG	RB

Table 3: The table shows the different kinds of calls based on the initial estimation and the final outcome. l_α denotes an admission threshold applied after a probing interval, while l_a is the desired upper bound on the loss level.

If we return to the strict policy introduced above, it minimizes class AB while class RG is large, thus protecting on–going calls in a more successful manner whilst increasing the blocking probability. In the same way, the relaxed policy minimizes class RG, thus reducing the blocking probability at the risk of a higher number of bad calls.

To obtain absolute numbers on the number of calls in the classes, a real loss distribution has to be considered. By aiming at an upper bound of the loss rate and applying the *cdf* to that bound, it is possible to get absolute numbers of the

different classes. The admission threshold can then be varied to find a desired optimum.

Figure 12 shows an example of a uniform loss distribution (calls can experience any packet loss rate between 0 and 100% with equal probability) with a desired upper bound on the loss rate l_a. The *cdf* of four seconds in Figure 10 has been superimposed[1] on the uniform loss distribution for two admission control policies, namely strict and relaxed. The number of calls belonging to each class can be determined by the areas in the graph. The areas are bounded by l_α and the *cdf*. For example, it can be seen from the graph that area RG (rejected calls that turned out good) is large in the strict policy, but is small in the relaxed. Likewise, area AB (admitted calls that turned out bad) is small in the strict and large in the relaxed policies.

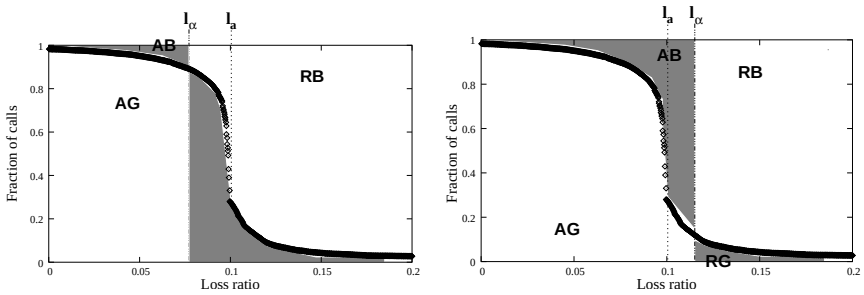

Fig. 12: Example showing the result of imposing admission control decision in the strict (left) and the relaxed (right) admission policy with a uniform loss distribution. The desired upper bound on packet loss is l_a and the imposed threshold is l_α.

A uniform loss distribution is evidently unrealistic, but the same methodology can be applied for a real loss distribution. We have applied the method to the complete set of 9683 error-free calls in the measurements in the case of four seconds probing and calculated the percentage of calls that fall in each of the areas. The rest of this section deals with this case. Figure 13 shows the blocking probability (RG+RB) for the complete sample space. From the figure it can be seen that rejecting calls that experience an initial loss rate equal or higher to 10% gives a blocking probability of around 15%, whilst a more stringent packet loss rate threshold would result in a rapidly increasing blocking probability. Note however, since the error-free calls are omitted, the blocking probability is overly pessimistic. We would expect a lower blocking probability with a factor of around three if the error free calls were actually included.

As was previously noted, the accuracy of an admission control policy can be measured by counting the correct and incorrect decisions. Figure 14 shows the incorrect decisions for a packet loss rate target of 2%. The plot shows both kinds (AB and RG) as well as their sum. The plot illustrates how the number

[1] Note, the *cdf* is reflected around x=0.

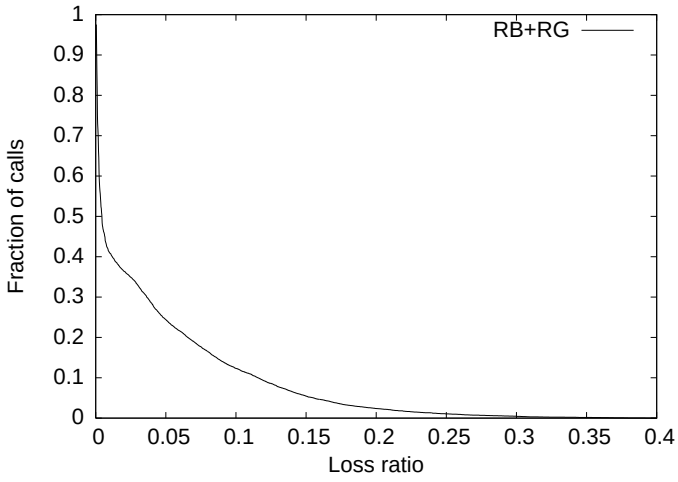

Fig. 13: Blocking probability as a function of the packet loss rate admission threshold.

of incorrectly admitted calls increases as the admission threshold is relaxed, while the incorrectly rejected calls decreases. The sum of the two functions has a minimum for a particular admission threshold at 1.8%, which can be considered as an optimum operating point. That is, a minimal number of incorrect decisions were made at this threshold. Finally, Figure 15 illustrates the sum of incorrect decisions for different target loss rates as the acceptance threshold is varied. The results show a minimum close to the value of the target loss rate, as was intuitively expected.

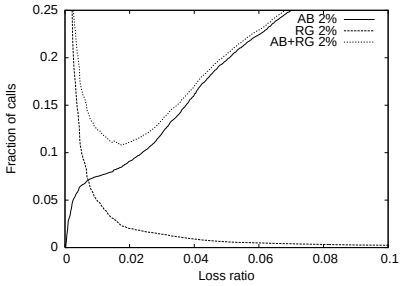

Fig. 14: Erroneous decisions as a function of the admission threshold for a 2% target loss rate

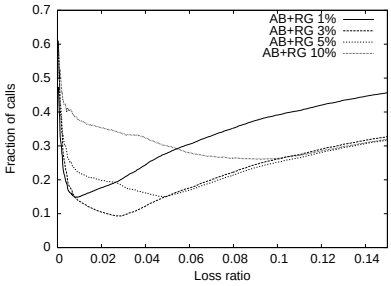

Fig. 15: Erroneous decisions as a function of the admission threshold for different target loss rates.

The choice of an operating point for the admission control has to take into account many parameters. We can always increase the accuracy by measuring

over a longer period. However, increasing the probing period reduces the advantages, since we are extending the period in which a bad call is disturbing the ongoing calls, reducing the overall quality in the process. Also, longer probing times increase possible frustration in the case of a rejection.

To summarize, if we use our measurements, we would probe for four seconds and use an admission threshold close to the targeted value. Assume that 2% packet loss is acceptable. In this case, the admission threshold should also be around 2%, which would give a blocking probability of 36%. The admission control decision would then have failed 11% of the time, the majority would be calls that were admitted although they turned out to be bad (9% of the total calls), a smaller fraction would be calls that were rejected but turned out to be good (2%).

4 Conclusions and future work

This paper proposes a quality differentiation scheme based on self-admission control without the need of infrastructure changes. The admission control is performed at the application layer and can provide statistical bounds on the packet loss rate that stream flows will experience in the network. We have shown how the admission control mechanism can be devised by blocking calls experiencing an initial loss rate exceeding an admission threshold. An initial admission threshold is motivated by two factors: (1) it makes sense to drop calls that will experience bad quality and thus reduce congestion in the network so that other calls may experience better quality; (2) an audio codec will have an upper bound on quality: exceeding a drop rate will result in unacceptable audio quality.

We have evaluated the admission control scheme by analyzing a large number of IP telephony calls that were made over the Internet. Based on this empirical data, we have shown that it is possible to predict the quality of a call by making an early measurement of the packet loss. From our particular data, we have shown that it is sufficient to make an estimation after four seconds. The analysis we have performed offers thresholds for call blocking probability and failure rates of the scheme.

From a practical point of view, the admission control scheme shown in the paper could be implemented using standard RTCP [SCFJ96] receiver reports. A small adjustment of the rate that the receiver generates the reports would be enough for our probe-based admission control scheme.

One limitation with our method is that all calls in the experimental data are in fact admitted. The effects of dropping calls to the network as a whole has not been assessed. We claim that this observation is irrelevant in this study for two reasons: (1) all of the calls in the study were disjoint in time; (2) the effect could only be positive, thus our results can be seen as a worst-case.

An interesting point is whether the results based on the measured data [ML03] are generally valid. This is a difficult question, and we cannot claim that the results hold for all network conditions. For example, one could claim differences in timescales (the measurements were made in 2001), networks (most data were

made on academic networks), link technologies (no wireless access were available, etc). We hope that future work can obtain a better understanding of such conditions.

References

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W Weiss. An architecture for differentiated services. RFC 2475, IETF, December 1998.

[BHK+03] Pravesh Biyani, Olof Hagsand, Gunnar Karlsson, Ian Marsh, and Ignacio Más. Early estimation of voice over ip quality. In *Proc. of the 21st Nordunet network conference*, Reykjavik, Iceland, August 2003.

[BKS+00] Lee Breslau, Edward W. Knightly, Scott Shenker, Ion Stoica, and Hui Zhang. Endpoint admission control: Architectural issues and performance. In *Computer Communication Review – Proc. of Sigcomm 2000*, volume 30, pages 57–69, Stockholm, Sweden, September 2000.

[FKR00] Viktoria Fodor, Gunnar Karlsson, and Robert Rönngren. Admission control based on end-to-end measurements. In *Proc. of the 19th Infocom*, pages 623–630, Tel Aviv, Israel, March 2000.

[HMH03] Olof Hagsand, Ian Marsh, and Kjell Hansson. Sicsophone: A low-delay internet telephony tool. In *Proc. of the 29th Euromicro Conference*, pages 189–197, Belek-Anatalya, Turkey, September 2003.

[Kar98] Gunnar Karlsson. Providing quality for internet video services. In *Proc. of CNIT/IEEE ITWoDC 98*, pages 133–146, Ischia, Italy, September 1998.

[MFK01] Ignacio Más, Viktria Fodor, and Gunnar Karlsson. The performance of endpoint admission control based on packet loss. In *Procedings of the 2nd COST 263 International Workshop on Quality of Future Internet Services*, volume 2156 of *LNCS*, Coimbra, Portugal, September 2001.

[MIK01] Ignacio Más Ivars and Gunnar Karlsson. PBAC: Probe–based admission control. In *Procedings of the 2nd COST 263 International Workshop on Quality of Future Internet Services*, volume 2156 of *LNCS*, pages 97–109, Coimbra, Portugal, September 2001.

[ML03] Ian Marsh and Fengyi Li. Wide Area Measurements of VoIP Quality. In *Procedings of the 4th COST 263 International Workshop on Quality of Future Internet Services*, volume 2856 of *LNCS*, Stockholm, Sweden, October 2003. Springer.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real–time applications. RFC 1889, IETF, January 1996.

[SRC84] Jerome H. Saltzer, David P. Reed, and David D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, November 1984.

Paper G

Ian Marsh, Juan Carlos Martín Severiano, Victor Yuri Diogo Nunes and Gerald Q. Maguire Jr.
IEEE 802.11b voice quality assessment using cross-layer information. In *1st Workshop on Multimedia over Wireless (MediaWIN)*, Athens, Greece, April 2006.

"One day you'll walk right out of this life,
And then you'll wonder why you didn't try"

Ghosts - The Jam

IEEE 802.11b voice quality assessment using cross-layer information

Ian Marsh[1], Juan Carlos Martín Severiano[2], Victor Yuri Diogo Nunes[2], and Gerald Q. Maguire Jr.[2]

[1] CNA, SICS, Sweden
ianm@sics.se
[2] School of Information and Communication Technology, KTH, Sweden
juancarlosmartin@gmail.com, ynunes@gmail.com, maguire@kth.se

Abstract. This paper reports on the suitability of IEEE 802.11b networks for carrying real-time voice traffic, considering particularly the end terminals. More specifically we have looked at 802.11 networks in different operating circumstances: an outdoor environment, an office environment, and the impact of competing traffic for real-time voice. Additionally we have investigated the link layer operation together with the application layer. Based on over 2500 recorded sessions, it can be generally concluded that the 802.11b technology can support real-time voice; particularly if the link transmission rate is immediately lowered after an unsuccessful initial transmission. However, we did find situations where the voice quality deteriorated below commonly accepted values, such as when competing with high-rate TCP traffic, when intervening obstacles blocked the transmission path, and with some cases using the RTS/CTS mechanism.

1 Introduction

IEEE 802.11b networks are being used in public hotspots, along with office and home networks. The resulting broadband wireless local area network (WLAN) has brought IP-based telephony into competition with the cellular telephony infrastructure. The goal of this paper is to assess the suitability of 802.11 networks based on extensive measurements. Our focus was the *application* perceived quality in different usage scenarios due to the environment's effect on voice quality. By environment we mean the physical context: the separation of the nodes and intervening obstacles. We focus on the quality variations of a single voice over IP (VoIP) call in various circumstances: in ad-hoc mode with clear line of sight, inside and outside an office environment, in the presence of competing TCP and UDP traffic, and using 802.11 infrastructure mode (i.e. with an access point). Where possible, we also examine the contribution of the MAC layer, specifically retransmissions and the RTS/CTS mechanism with the voice application. The remainder of this paper is structured as follows: the next section explains our basic experimental setup, and how one can use information from two layers to obtain data about the overall VoIP quality. Section 3 presents some of the related

work in this area, Section 4 gives our findings in the four measurement settings. Finally we conclude with a summary of our findings and highlight issues for 802.11 real-time voice users together with some suggestions for future work.

2 Basic experimental configuration and cross-layer measurement

The basic configuration used for our experiments comprises one node that sends a unidirectional flow of RTP VoIP packets to a second node that acts as a receiver. The basic configuration we used is shown in figure 1.

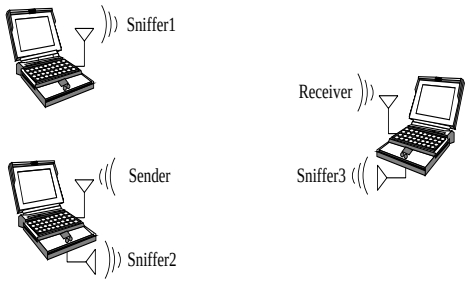

Fig. 1. VoIP measurement testbed

In order to observe and capture over the air traffic, we used two monitoring devices placed close to the sender station. One monitor was in the same sending station and one was physically separate[1], shown as sniffer 2. Capturing all the frames in the air can be problematic, and our decision to use two capture devices and merge the traffic they observe was motivated by the experience and pitfalls reported by Yeo, et al. [YBA02]. We also captured the traffic sent back from the receiver using a third monitor, sniffer3, however this will be mostly ACK frames.

Ethereal and Sphone were used to capture the link layer frames and VoIP packets respectively. During the experiments, the RTP traffic corresponded to a simulated call of 80 seconds generating 160 byte packets 20ms apart, no silence suppression or signaling were used. As an example of combining cross-layer information Figure 1 shows the retransmission pattern of 4000 packets ($1/20\text{ms} \cdot 80$ seconds) when observed at both the data link layer and the application layer for a single VoIP flow. Vertical lines up to $y = 3$ show the number of transmissions that were received. In some cases the 4^{th} transmission (i.e. $y = 4$) were not received and are shown as loss in the RTP stream (circles in the plot). The default maximum number of transmission attempts by the MAC layer was four with the hardware we used, after which the frame is silently discarded. Note that only by using the application layer information at the receiver was it possible to

[1] Experiments showed the probability of both monitors losing a frame was 0.04%.

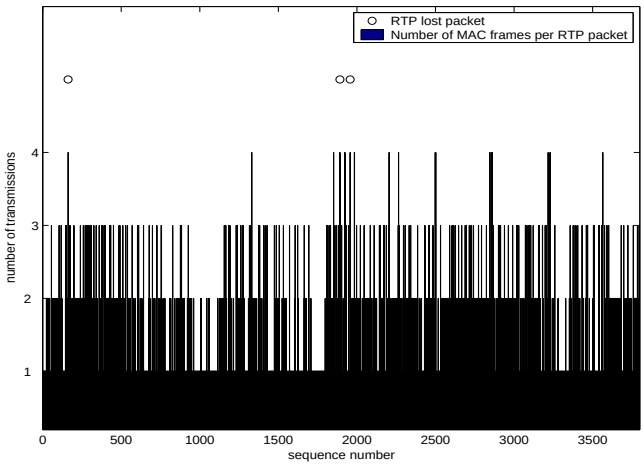

Fig. 2. Typical data link layer behavior

establish whether the transmission was actually received or not. This is because the ACK may not always be received, and failed link layer transmissions are not immediately indicated by the drivers to the higher layers.

3 Related work

Anastasi et al. measured the performance of IEEE 802.11b ad-hoc networks [GAEB04], specifically the range of the end-terminals, the impact of different data rates and their variability. They observed that the transmission range was highly dependent on the data rate up to 100m, whilst the physical carrier sensing range was independent of rates up to 200m. Unlike their results in ad-hoc mode, we didn't observe different rates up to 320 meters. Even at 400 meters there was no conclusive data rate dependency on range.

Hertrich looked at mixed traffic (including real-time voice) in IEEE 802.11 networks [Her03]. He used a MAC booster and by tailoring it, could alter the number of retransmissions at different positions to achieve the required throughput. We did not try to change the number of transmissions. This work is similar to ours in that he considered the environment as important, however he used VoIP and MPEG4, while we used VoIP. Additionally, Hertrich focused on the home, whereas we focused on an office environment. Dimitrou et al. address issues that can make the deployment of multimedia communications difficult in 802.11 networks [DS03]. They cite interference and users moving out of range as limiting factors for good VoIP quality in 802.11 networks. They suggest the use of smart speech coding (including an enhanced version of the G.711 coding developed by their company) to make the speech more resilient to loss.

Hoene et al. examined the effect of motion on the performance of wireless links through a series of experiments with moving nodes [HGW03]. They con-

clude that other factors such as modulation type, quality of power supply, environmental setup, and number of retransmissions may have greater impact on 802.11b performance than the motion itself. In general the greater the speed of the terminal the lower the correlation of loss events. In our experiments the nodes were not moving, i.e. movement only occurred between measurements, thus movement should only decrease the observed losses.

4 Results

We will now present four distinct series of experiments regarding VoIP quality: (A) outdoor measurements, (B) measurements in an office, (C) the effect of competing traffic, and (D) the usage of the RTS/CTS mechanism.

4.1 Quality as a function of distance

The first measurement series we conducted were designed to examine the effect of the distance between the sender and the receiver when using ad-hoc mode. The terminals were within line of sight in an outdoor environment. Figure 3 shows the averages of transmitting a single VoIP flow eight times at each distance. It is a histogram of the percentage of MAC frames successfully delivered at the seven different distances.

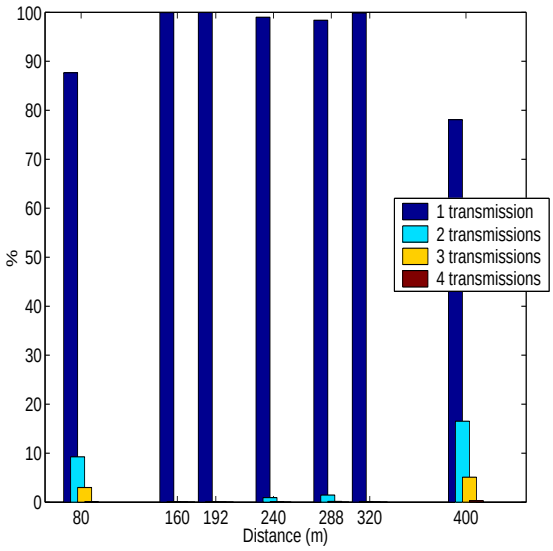

Fig. 3. Number of successful MAC transmissions vs distance

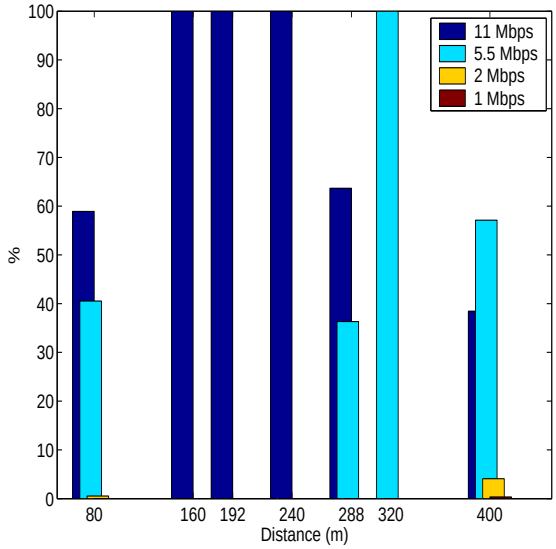

Fig. 4. IEEE 802.11b rates used vs. distance

Figure 4 shows the rates at which each of these frames were transmitted. The first observation is that the majority of transmissions were successful at the first attempt. This is particularly true for the middle distances in our measurements. Overall the loss percentages were either zero or very low ($< 0.1\%$), even considering the retransmissions at 80 and 400 meters. These were 0.025% at 80 meters and 0.05% at 400 meters. The loss and jitter figures can be found in Juan Carlos Martín Severiano's master thesis, along with further details and analyses [Sev04]. It is evident from these figures that interference from nearby networks operating at the same or similar frequencies can induce losses even at relatively short distances. One effect of this is the lower rates as easily seen in Figure 4. No continual competing traffic was observed on the channel during these measurements, hence the delay and jitter values are low ($<$ 7ms for both). Interference on the same channel was the result of IEEE 802.11 beacon frames and probe requests from (assumed) nearby 802.11 networks.

4.2 Office environment measurements

Next we measured the number of MAC layer transmissions inside a typical modern office environment. The purpose of this scenario was to measure the effect of the walls, windows, and intervening obstacles typically found in modern offices on the 802.11 transmissions. The floor-plan is shown in figure 6 and the actual environment in figure 7. The letters on the floor-plan reflect the positions of the sender (O_1 and A), the receivers (the remaining letters) and orientations of

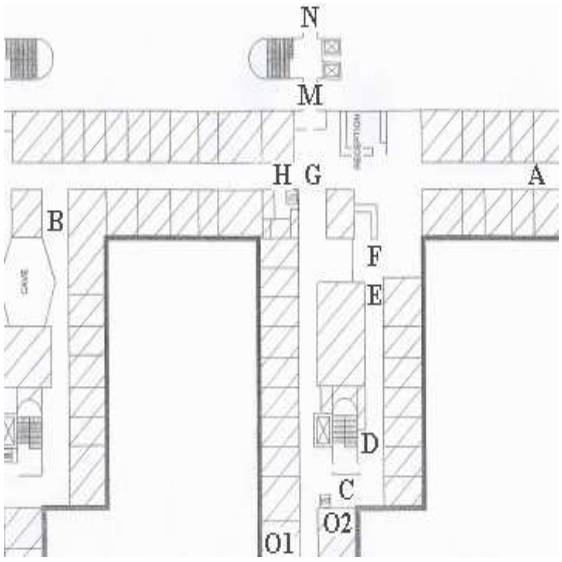

Fig. 5. The office floor-plan for the indoor experiments

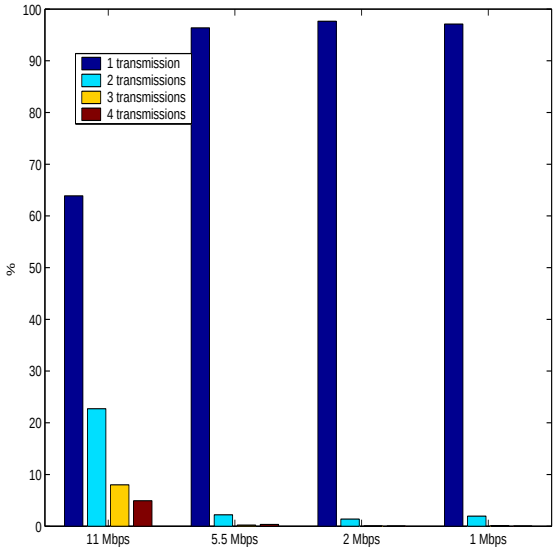

Fig. 6. The bitrate histograms for the indoor experiments

the receiver (indicated as numerals). The placements of the receiver were chosen to represent challenging locations for wireless communication, for example a computer room (containing servers) is located between O_1 and E, F. With the sender located at A and the receiver located at B, we exam at the impact of the office environment on voice quality.

Fig. 7. Office environment with three different receiver (and antenna) orientations

Figure 6 shows the mean of four separate series of measurements where the transmission rate was fixed at each of the defined rates for an 802.11b interface. Unlike the example presented in section 4.1 where the rate could vary, we observed that using a fixed sending rate of 11Mbits/sec led to many more retransmissions and higher losses, approximately 2.75% of the total. By reducing the rate, the probability of a successful first transmission increases, as can be seen from the 5.5Mbits/sec values. A clear conclusion is that for voice traffic the rate should be immediately reduced (for the retransmission), rather than attempting retransmissions at a higher rate. This may not only alleviate loss, but reduce delay and jitter for this frame[2]. The loss rates for the 5.5, 2, and 1 Mbits/second rates were approximately 1%.

From table 1 we can see that the environment significantly affects the quality, particularly in the larger loss values, in some situations communication was not even possible (F_2). Different locations and antenna orientations around the office were selected as shown in figures 6 and 7 respectively. From Table 1 the quality differences were significantly affected by the antenna orientation. For each of the orientations there is 90 degree difference in the X, Y, and Z planes, indicated by the 1,2 and 3 subscripts in the table. Further details of the application layer measurements, particularly within the office environment can be found in [Nun04].

4.3 Competing UDP and TCP traffic in ad-hoc mode

We now look at the effect of competing traffic on the quality of VoIP sessions. We first consider the case of ad-hoc mode, i.e. without an access point. The

[2] This confirms a hypothesis by one of the authors (GQMjr) presented to Andrzej Duda following his talk at KTH on 2003.05.08.

Pos.	Loss (%)	RTT (ms)	Jitter (ms)
O_2	[0, 0.0, 0.1]	[1.9, 2.2, 2.4]	[0.2, 0.2, 0.3]
C	[0, 0, 0]	[1.9, 2.1, 4.0]	[0.1, 0.1, 0.2]
D	[0, 0, 0]	[2.1, 2.6, 3.1]	[0.1, 0.7, 1.0]
E_1	[0, 0.2, 2.6]	[2.8, 3.2, 5.4]	[1.0, 1.2, 2.0]
E_2	[9.4, 36.3, 89.1]	[5.3, 12.2, 24.3]	[4.2, 6.3, 16.1]
E_3	[0, 0.0, 0.2]	[2.8, 2.8, 4.0]	[0.9, 1.1, 1.5]
F_1	[20.1, 54.9, 88.7]	[5.4, 13.4, 24.6]	[0.9, 14.2, 57.9]
F_2	No signal	No signal	No signal
F_3	[1.8, 22.8, 84.9]	[4.6, 11.7, 13.7]	[2.9, 6.7, 37.4]
G	[0, 0.2, 1.9]	[1.9, 2.2, 3.8]	[0.1, 0.4, 1.0]
H_1	[0.4, 5.4, 30.2]	[3.5, 3.9, 8.1]	[1.5, 2.2, 4.2]
H_2	[3.4, 11.0, 28.3]	[5.9, 6.1, 11.7]	[3.4, 3.9, 4.6]
H_3	[0, 0.2, 2.7]	[3.4, 3.4, 5.4]	[1.1, 1.2, 1.8]

Table 1. VoIP metrics in an office environment ([min,mean, max])

two VoIP nodes (and monitors) were in the same room with up to four nodes generating UDP or TCP background traffic. The TCP and UDP packets were 1500 bytes in total, produced by the NTTCP traffic generator. Each node was responsible for generating a single stream. The goal was to observe the MAC protocol's behavior by measuring the delay, jitter, and loss caused by the failure of 802.11's collision avoidance mechanism under increasing load.

If stations select identical slot numbers and hence send simultaneously, collisions will occur, resulting in lost frames. Usually the packet capture nodes cannot detect collisions so it is the responsibility of the application layer to infer lost packets due to heavy load on the medium. Usage of UDP was intended as a controlled traffic source, and TCP as a more representative, but more complex, traffic source. Figures 8 and 9 show the round trip time and jitter for the configuration described. This RTT is calculated from application layer RTCP information sent once per second, hence the high variance in the measurements.

Zero nodes in the figure indicates the case without competing traffic, i.e. solely the VoIP flow. For one or more competing nodes it is noticeable that the delay and jitter values are much higher compared to the ones shown earlier. Since delay should be less than 150ms for good interactive communication, a significant proportion of the delay can be used up gaining channel access. Note also the standard deviation of the TCP traffic is much higher than the UDP, whereas the mean is only slightly higher. This variance is problematic for the jitter of VoIP sessions, leading to either loss or further delay due to the buffer playout algorithms.

4.4 Competing TCP traffic in infrastructure mode

Next we examined the VoIP quality when using an access point between the communicating nodes, as this is the most common scenario for 802.11 wireless (voice) access to the Internet. We only considered TCP traffic as a competing

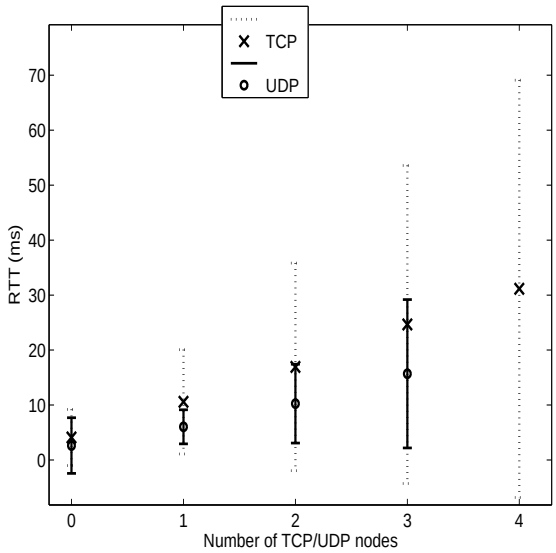

Fig. 8. The round trip time for 0-4 competing nodes in ad-hoc mode

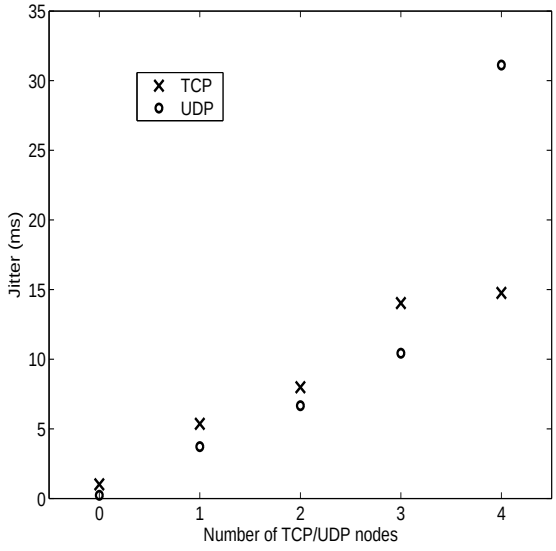

Fig. 9. The jitter for 0-4 competing nodes in ad-hoc mode

traffic source for these experiments. The access point (AP) used was a D-link DI-614+, which is one of the most popular commercial APs. Due to the limited number of stations used, we decided to configure the nodes to send at full rate however, in reality, a larger number of smaller TCP flows from several users would compete on the same 802.11 channel.

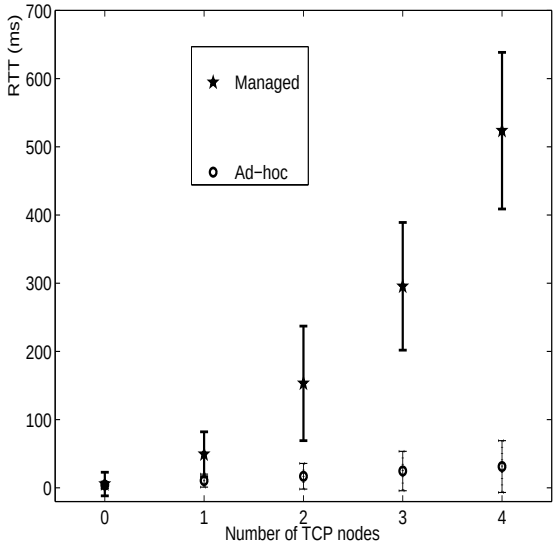

Fig. 10. The round trip time for 0-4 competing nodes

In figures 10 and 11 we show the round trip time and jitter for zero to four competing nodes. One possible explanation of the higher delay in infrastructure mode is internal scheduling/queuing within the access point, the frame also must be transmitted on the medium twice, and there is the effect of TCP's congestion avoidance mechanism [HRBSD03]. Alternatively the AP has the same probability of accessing the media as any other station, the rise in the delay could be simply due to the increased number of transmitting nodes. What is clear is that this situation can lead to delays that would be unacceptable for VoIP users, far exceeding the delay for good interactivity. We observed some loss, but it was low and within the acceptable values for VoIP quality for the speech coding scheme we used (G.711). A detailed examination of the performance of access points under heavy load can be found in [Pel04].

4.5 RTS/CTS mechanisms for VoIP performance

In this set of experiments we wanted to ascertain if the RTS/CTS handshake mechanism is effective against the hidden node problem. For this purpose we

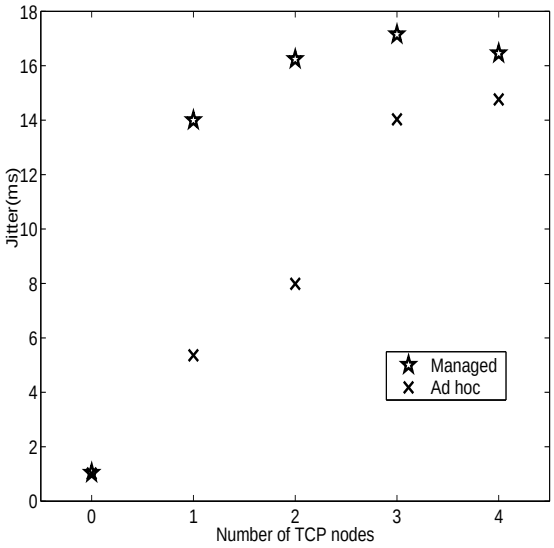

Fig. 11. The jitter for 0-4 competing nodes

placed the sender and receiver stations out of transmission range of two additional stations, so that each pair of stations were hidden from each other, as shown in Figure 12. Normal office walls were the obstacles attenuating the radio signals at the hidden stations. At the intersection of these hallways we placed an access point, which was in range of every wireless stations. We then examined six combinations of using the RTS/CTS mechanism:

A No background traffic, RTS disabled in sender station
B No background traffic, RTS enabled in sender station
C Two stations sending TCP, RTS disabled in all the stations
D Two stations sending TCP, RTS enabled in all the stations
E Two stations sending TCP, RTS enabled in voice sender station only
F Two stations sending TCP, RTS disabled, stations not hidden

Figure 13 shows the loss results. We will begin by considering the experiments without background traffic, i.e. A and B. The plot shows 3% of losses when the RTS mechanism is used.

By adding background traffic, via enabling the TCP flows between nodes hidden from the sender and receiver stations, we observed that the hidden nodes caused a 25% loss in the voice stream (the C case), whilst the loss percentage was only 0.3% when the stations were not hidden (F). In order to reduce the loss we enabled the RTS mechanism in all the stations, and the loss percentage reduced to approximately 2% (D). We also examined whether enabling RTS in the voice sender station alone would help to minimize loss. However, rather than reducing loss it increased it up to 50% (E). This shows that the RTS

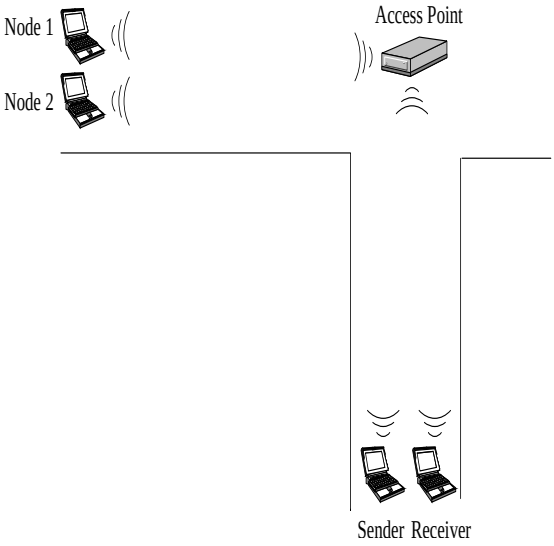

Fig. 12. A) RTS/CTS setup

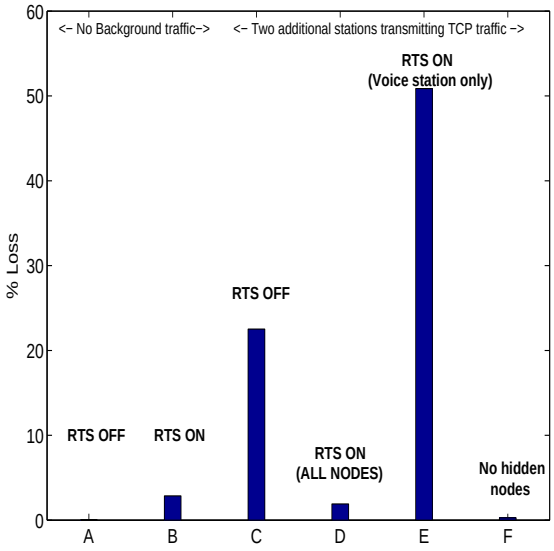

Fig. 13. B) RTS/CTS setup

mechanism is effective against the collisions caused by hidden nodes, but only if *every* station enables it. In fact in our experiments enabling it in only one station causes greater degradation than leaving it disabled.

Experiment	A	B	C	D	E	F
Throughput (Mbps)	-	-	1.1	1.1	1.56	2.1

Table 2. Throughput for different configurations of the RTS/CTS mechanism

Clearly, there was no decrease in aggregate throughput after enabling RTS in all the nodes. However, the throughput increased when the voice station was the only one that enabled RTS. Thus the RTS mechanism benefited the other stations whilst the enabling station experienced worse performance. The highest throughput was achieved when all the nodes were in range, because the probability of collision was lower. Figures 14 and 15 show the delay and jitter therefore. Here we observe the overhead introduced by the RTS mechanism can be significant. The round-trip delays obtained in scenario F were very high, approximately 1.5 seconds for the RTT. Other than software errors we have no explanation for this high delay. We conclude from these measurements that the RTS/CTS mechanism is not beneficial to VoIP users unless *all* stations enable it. Since it is disabled by default it is safer not to enable it optimistically.

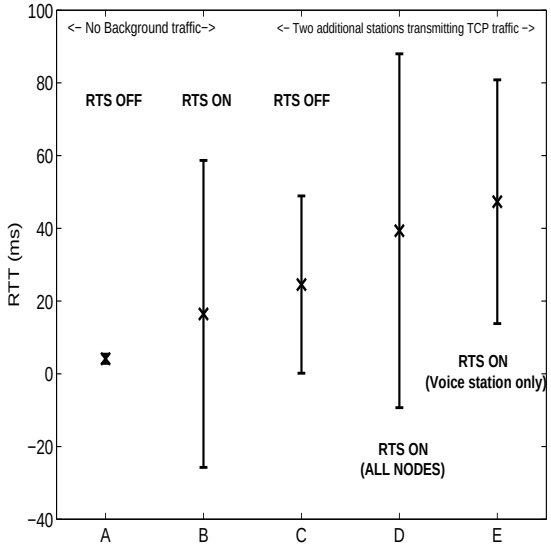

Fig. 14. RTS/CTS delays

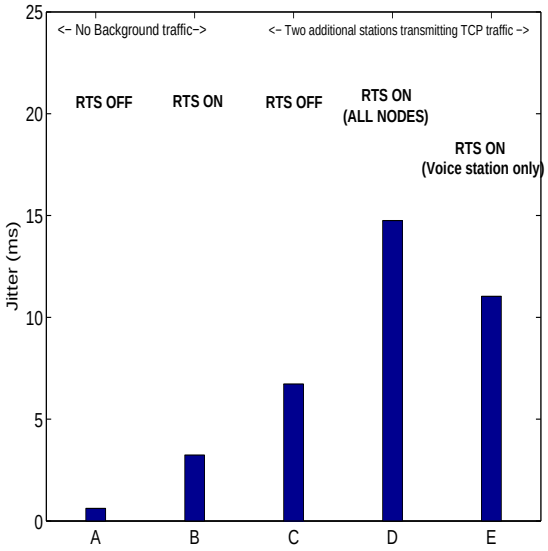

Fig. 15. RTS/CTS jitter values

5 Conclusions

We have conducted many hundreds of experiments in order to assess the suitability of 802.11b networks for real-time voice communication. We have found that measuring the MAC layer behavior in *conjunction* with measuring the application layer performance is both useful and informative in estimating the quality of VoIP sessions. It is informative in that the occurrence of retransmissions, for example, can indicate that the terminal is entering/experiencing a period of poor quality.

The contribution of the MAC layer itself is generally low, however delay, once introduced into a system, cannot be eliminated, unlike the perceptual effects of loss for example, so understanding the MAC layer's contribution, particularly in terms of delay is important. It is situation-specific as to whether 802.11b can deliver sufficient real-time voice quality. The major hurdles we encountered were attenuating objects between the end-terminals.

We have also seen that (at least) one popular access point can add delays that would seriously degrade of conversations under heavy load. Although the newer standards 802.11{a,g} allow greater capacity, they operate over shorter ranges. In the case of competing traffic, 802.11e will give priority to voice traffic when in competition with TCP bulk transfers, therefore we would recommend its use based on the findings from this work.

References

[DS03] Eleftherios Dimitriou and Patrik Sörqvist. Internet Telephony over WLANS. Technical report, Global IP Sound, Sep 2003. http://www.globalipsound.com/solutions/wlan.usta_paper.pdf.

[GAEB04] Enrico Gregori Giuseppe Anastasi Eleonora Borgia, Marco Conti. IEEE 802.11b Ad Hoc Networks: Performance Measurements. In *Cluster Computing*, 2004.

[Her03] Daniel Hertrich. Experimental Performance Evaluation of an 802.11b WLAN supporting mixed Multimedia Traffic. Technical report, TU Berlin, Sep 2003.

[HGW03] Christian Hoene, André Günther, and Adam Wolisz. Measuring the Impact of Slow User Motion on Packet Loss and Delay over IEEE 802.11b Wireless Links - Row Measurement Traces. In *Proceedings of Workshop on Wireless Local Networks (WLN) 2003*, Bonn, Germany, October 2003.

[HRBSD03] Martin Heusse, Franck Rousseau, Gilles Berger-Sabbatel, and Andrzej Duda. Performance anomaly of 802.11b. In *In Proceedings of IEEE INFOCOM 2003*, San Francisco, USA, Mar 2003.

[Nun04] Victor Yuri Diogo Nunes. VoIP quality aspects in 802.11b networks. Master's thesis, IMIT, Royal Institute of Technology, Stockholm, Sweden, August 2004.

[Pel04] Enrico Pelletta. Maximum Throughput of IEEE 802.11 Access Points: Test Procedure and Measurements. Master's thesis, IMIT, Royal Institute of Technology, Stockholm, Sweden, May 2004.

[Sev04] Juan Carlos Martín Severiano. IEEE 802.11b MAC layer's influence on VoIP quality: Measurements and Analysis. Master's thesis, IMIT, Royal Institute of Technology, Stockholm, Sweden, October 2004.

[YBA02] Jihwang Yeo, Suman Banerjee, and Ashok Agrawala. Measuring Traffic on the Wireless Medium: Experience and Pitfalls. Technical report, Department of Computer Science University of Maryland, Dec 2002.

Paper H

Ian Marsh, Björn Grönvall and Florian Hammer.
The design and implementation of a quality-based handover trigger. In *5th IFIP-TC6 Networking 2006*, Coimbra, Portugal, May 2006.

"Tell me where it hurts
to hell with everybody else
All I care about is you and that's the truth
They don't love me; I can tell
But you do, so they can go to hell"

Tell me where it hurts - Garbage

The design and implementation of a quality-based handover trigger

Ian Marsh[1], Björn Grönvall[1], and Florian Hammer[2]

[1] SICS, Kista, Sweden
ianm@sics.se, bg@sics.se
[2] Telecommunications Research Center (ftw.) Vienna, Austria
hammer@ftw.at

Abstract. Wireless connectivity is needed to bring IP-based telephony into serious competition with the existing cellular infrastructure. However it is well known that voice quality problems can occur when used with unlicensed spectrum technologies such as the popular IEEE 802.11 standards. The cellular infrastructure could provide alternative network access should users roam out of 802.11 coverage or if heavy traffic loads are encountered in the 802.11 cell. Therefore, our goal is to design a handover mechanism to switch ongoing calls to the cellular network when the 802.11 network cannot sustain sufficient call quality. We have investigated load and coverage scenarios and designed, implemented and evaluated the performance of an 802.11 quality-based trigger for the handover of voice calls to the cellular network. We show that our predictive solution addresses the coverage problem and evaluate it within a real setting.

1 Introduction

Handsets that are equipped with multiple standard radios will become commonplace. PDAs with 2G cellular radios and IEEE 802.11 chipsets are already on the market, and dual-radio mobile phones are also beginning to appear. The primary motivations for a voice handover system are monetary. By connecting to 802.11 access points when available, it should be possible to avoid cellular tariffs. However when users leave the 802.11 area they may want to continue their voice calls. Therefore a handover mechanism to alternative technologies for voice users is desirable. Excess traffic within an 802.11 cell is also a reason to handover a call to the cellular system. The basic problem is when to perform, or even schedule, a handover from one system to the other. The cellular infrastructure provides network support for its clients, and performs the handover on their behalf. The clients periodically report their reception status enabling the infrastructure to make an informed handover decision. In an 802.11 system this functionality is not available, therefore it becomes the task of the handset when best to handover a session. Prediction is the key issue with this approach as voice call setup takes approximately five seconds to the fixed or cellular network. This is an average value we observed by repeatedly calling to the PSTN and GSM

networks. During the handover, ideally no quality differences should be audible making the handover as transparent as possible. On the other hand, the system should not handover voice calls to the cellular system due to small audio glitches that many mobile users have become accustomed to, worse still switch back or forth between network types. Manual switching should always be an option, if users want to use the cellular network. However, in this work we assume that users want to use the 802.11 networks for voice communication when available. Therefore the contribution of this work is an *automatic* handover solution for real-time voice sessions on 802.11 networks to the cellular infrastructure when poor quality conditions persist.

2 Assessing the influence of packet loss using PESQ

Packet loss is critical when determining voice quality. Bursty losses are well known to be commonplace in wireless communication, and 802.11 networks are no exception. Therefore the goal of this first evaluation is to ascertain how many packets can be lost in a burst without significant reductions in the perceptual quality. We do not consider delay or jitter in this first phase, only packet losses.

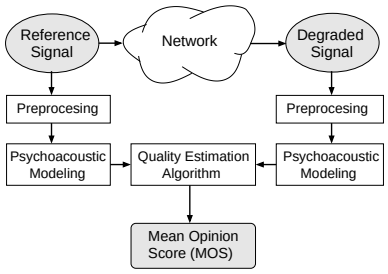

Fig. 1. The PESQ processing structure.

PESQ MOS	Linguistic equivalent	Quality degradation
4.5	Excellent	None
4	Good	
3.5	Good/Fair	Moderate
3	Fair	
2.5	Fair/Poor	Severe
2	Poor	
1	Bad	

Fig. 2. A quality degradation scale.

Figure 1 shows the functional units of PESQ, the ITU-T standard we derive our loss tolerances from [6]. A reference speech signal is transmitted through a network that results in a quality degradation corresponding to the path conditions and coding scheme. PESQ analyzes both the reference and degraded signal and calculates their representation in the perceptual domain based on a psychoacoustic model of the human auditory system. The disturbance between the original and the degraded speech signal is calculated by the quality estimation algorithm and a corresponding subjective Mean Opinion Score (MOS) is derived. The evaluation of speech quality using PESQ is performed off-line due to its computational complexity. For example a 400 packet sequence with ten losses requires approximately two seconds of processing time for simple G.711 coded speech. G.711 yields the maximum PESQ score (4.5) in the absence of loss, however it is particularly sensitive to packet loss even with concealment. We have

evaluated the tolerable loss lengths using both G.729 and iLBC, but they were always less than G.711, i.e. G.711 can be considered a worst-case codec. It is also the format used in our fully integrated solution, and thus allows us to directly set the loss thresholds in the handover trigger function without any transformation.

Figure 2 shows the PESQ MOS scale as defined by the ITU and their English linguistic equivalents. We have added an extra column, quality degradation, to indicate the quality reductions we have looked at as part of this first phase. The degradation of a MOS point is referred to as "moderate" and two points as "severe". We degrade the complete ITU-standardized eight second speech sample with 1 to 50 continuous losses. For each of the 50 loss bursts, we record the MOS score, and then shift the pattern through the eight second sample until it has been completely assessed for loss sensitivity. The technique and its effectiveness is fully described in [2]. Since the results are highly influenced by the performance of the packet loss concealment (PLC) algorithms, we conducted the tests with and without PLC. The loss concealment algorithm used was the one standardized by the ITU for G.711 called G.711i [5].

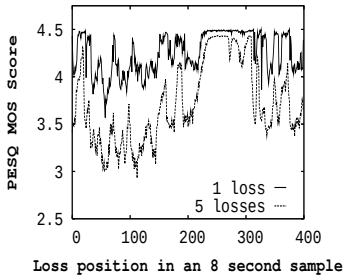

Fig. 3. Singular & quintuple loss scores for a female English speech sample.

Quality reduction	Gender	Language		
		English	French	Japanese
Moderate	Male	3/7	9/12	4/8
	Female	4/7	4/8	3/8
Severe	Male	30/31	43/45	45/46
	Female	31/32	46/48	45/48

Fig. 4. Packet loss lengths for 1 & 2 MOS reductions. The first value of the X/Y pair is without using PLC, the second is with PLC.

Examples of single and quintuple consecutive loss lengths with loss concealment are shown in Figure 3. The sample is one from the ITU standard database and is an American English female, the text is "She broke her new shoelace that day, the coffee stand is too high for the couch" and lasts for seven seconds. Observe that the concealment works well for one lost packet, however five consecutive losses are more difficult to conceal hence resulting in a lower PESQ score. Also note the silence period between samples 225 and 300 corresponding to the pause between the two phrases. The results for three different languages are given in Figure 4. The 90% percentile was taken for the MOS scores. As one can see the maximum number of consecutive packets one should allow in a burst without PLC is three for a moderate drop in quality for an English female speaker. However in reality loss concealment is employed in the receiver, in our full working system too, so we take seven as the threshold. It can be seen that English is the most sensitive amongst these three particular samples.

3 Emulating a mobile system

We move onto understanding the effect of other parameters on the design of a handover trigger by creating an experimental testbed. Our experiments have three major goals, first to gauge the impact of distance on wireless VoIP communication, second to understand the dynamics of voice streams mixed with TCP downstream traffic, and third how to measure and combine the available metrics suitable for implementing a handover trigger. Figure 5 shows the setup, it consists of a mobile terminal, a server we call a PBX, and load generating nodes. The PBX connects VoIP calls to the Public Switched Telephone Network (PSTN) and has the capability to handover calls to the cellular network when requested. The PBX and load generator are on a 100 Mbits/sec Ethernet, the mobile node and the sink are on the 802.11b network.

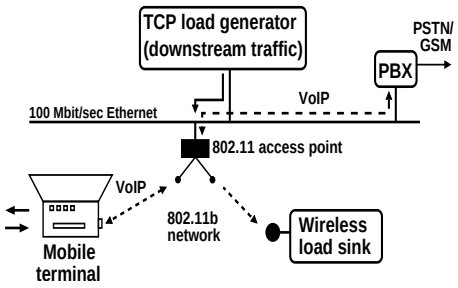

Fig. 5. The experimental testbed setup used in emulating a system capable of handover.

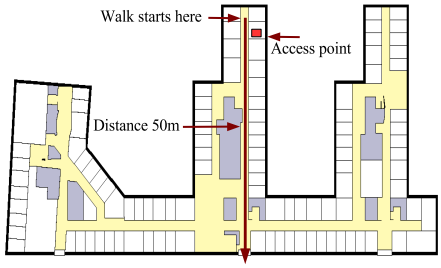

Fig. 6. The office layout used for our quality tests. The user walks straight out of the office.

The target network is expected to be used for voice applications, but also for traditional TCP-based applications such as email and web surfing. Therefore, we have developed a TCP NewReno load generator which attempts to create flows targeting a specified rate when network resources permit. For our stated goal of the design of a quality-based handover trigger, we will now explain three separate experiments:

Fading signal experiment: In this setup the mobile terminal moves past an access point and out of its coverage area. This is shown in Figure 6 as the arrowed line. The mobile terminal was carried along a corridor at walking speed and away from the access point. The left and center plots within Figure 7 show how voice packets arrive late or are lost due to environmental variations. From the figure we can see that during normal interference conditions there is little packet loss. As the signal deteriorates however, losses become much more frequent and the length of the loss bursts increase. One interesting observation is that packet losses occur much earlier at the mobile than at the PBX, compare the left and center plots. We assume this is due to better reception capabilities at the access point,

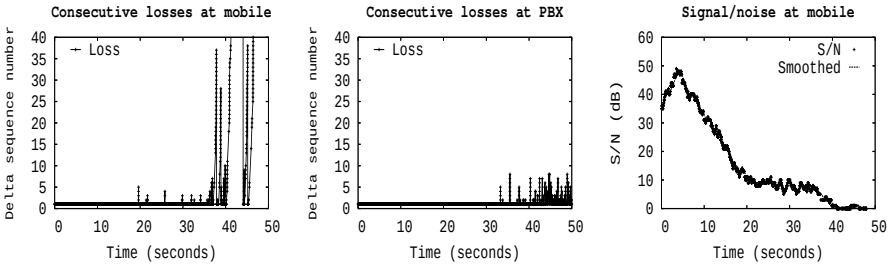

Fig. 7. Consecutive losses observed by the moving terminal (left) and the PBX (center) and signal strength as reported by the terminal (right).

for example better gain in the antennas or a dual antenna approach provides more diversity for receiving weak signals.

Packet losses are first experienced at the mobile, but in the target system it is the PBX that will perform the handover. This is because the functionality to handle both PSTN and IP calls is within the PBX. Therefore the PBX needs to be *continuously* monitoring the signal and network conditions at the mobile. This information can be sent either by piggybacking data onto the voice packets, or by sending RTCP-like designated packets at fixed time intervals as we do. From a system design perspective, it is critical that the PBX knows the state of the mobile.

The right plot of Figure 8 depicts how the mobile varies the transmission rate over time. During good signal conditions the mobile always uses the maximum transmission rate. During reasonable conditions the mobile varies the rate as it discovers link layer retransmissions become necessary [8, 9]. During poor conditions it constantly transmits at 1 Mbit/s. Notice that 1 Mbit/s is a *critical point*, as at this point it could lose connectivity altogether. Thus, when transmitting at 1 Mbit/s, a handover to the cellular network should be considered imminently, however it is not necessarily true that operating at 1 Mbit/s implies poor quality. Observe that a handover to the cellular network should ideally have completed at $t = 36$, which would have meant scheduling the handover approximately at $t = 31$ (the left plot of Figure 8), otherwise, poor quality could be experienced before the cellular call is in progress.

Loaded network: In this experiment we study the effects of a network operating close, but below, its full capacity. The synthetic load is limited to a target rate by our load generator. Due to the TCP behavior, the network will be overloaded for short periods of time. The synthetic load is directed towards (into) the 802.11 network in order to simulate web browsing or an email download. In this experiment we monitor an ongoing call and after ten seconds add synthetic load so that the network is operating at almost its full capacity. After a further ten seconds stop the synthetic load. In the left plot of Figure 9, we observe how the mobile at time $t = 10$ experiences a contiguous sequence of 13 packets that

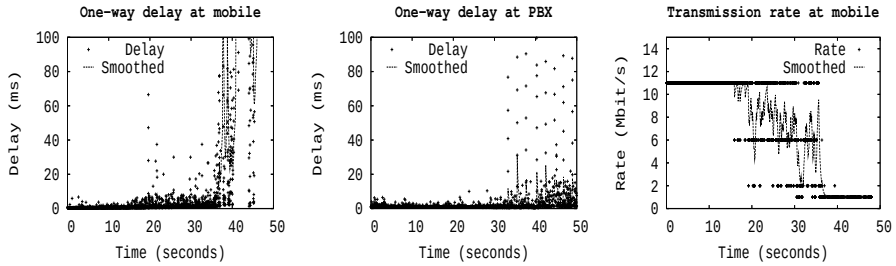

Fig. 8. Delays at the moving terminal (left) and the PBX (center) and changing transmission rates recorded at the terminal (right).

are delayed for more than 20ms and are effectively lost. This is because we used a constant size jitter buffer of 20ms in both the terminal and PBX.

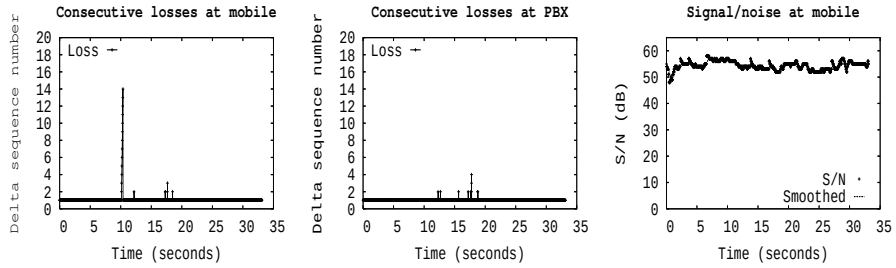

Fig. 9. Losses recorded at a stationary terminal (left) and a stationary PBX (center) shown with signal strengths (right) on a **loaded** network.

At the same time, we can see in the left graph of Figure 10 how the delay increases for each packet on its way from the PBX to the mobile, a queue is building up in the access point. The web servers are sending more packets into the 802.11 network than it can handle, and it takes time before TCP reacts and consequently backs off. During this time a queue builds up as packets arrive on the fixed network and they must be enqueued before gaining access to the congested 802.11 network. Since voice packets are delayed behind the TCP packets, they will eventually arrive late at the mobile. From the center graph of Figure 10, we can see how the delay from the mobile terminal towards the PBX increases when the network is loaded. The increase in delay is a result of the 802.11 contention, however in this case there is no extra queuing in the access point as the 100 Mbits/sec Ethernet is much faster than the 11 Mbits/sec 802.11b network. The asymmetry in the network speeds is clearly evident in these two cases. To conclude, we observe that loss events in either direction are rare, even in a loaded

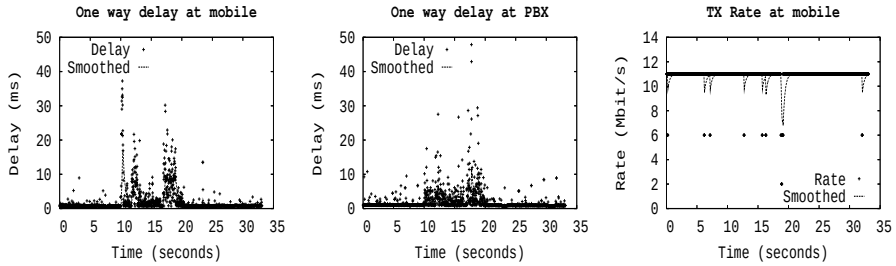

Fig. 10. Delays recorded at a terminal (left) and a stationary PBX (center) shown with transmission rates (right) on a **loaded** network.

network. For these loss events, the burst-loss length is typically one, and these can be dealt with using standard loss concealment methods such as G.711i.

Overloaded network: In a continuation of the previous experiment, but with a synthetic load driving the network to its maximum operating capacity. These figures are not included in the interests of space, but are briefly described. In these experiments, we observe serious loss problems from the PBX to the mobile, but not in the reverse direction, i.e. from the mobile to the PBX. This is to be expected, as we are again observing a queue building up in the access points as TCP packets arrive. It is trivial for the mobile to detect this and promptly inform the PBX since the traffic from the terminal to the access point is still unhindered. The problem arises with the speed this can occur. The network load can increase from unloaded to full capacity in a fraction of a second, however as we know it takes several seconds to establish a call through the PSTN. A better solution than triggering a handover in this case, is to give the voice traffic higher priority e.g. by marking the speech packets as having priority as proposed by the IETF Differentiated Services framework, for example by using the Expedited Forwarding (EF) class of service. The access point must also be capable of detecting these and scheduling the appropriate priorities.

4 Handover design and implementation

We now consider our real system with voice-enabled PDA's using commercial software, firmware and hardware solutions. When using real systems, the availability, reliability and resolution of network and link layer metrics are not the same on all systems. Therefore we chose not rely on one or two metrics rather to use a linear combination of those available for our trigger mechanism. Ideally we would like to use as many as possible *and* reliable, but certain hardware and software limitations prohibit this. The advantage of using this kind of combination is if the value is not available or reliable it contributes nothing, i.e. 0 to the overall score. The single value to make the handover decision we refer to as the **handover score**. The usable metrics we call the **handover contributors** and rationalize their inclusion in the following paragraphs. The scores are

derived from numerous experimental and empirical investigations as previously described.

Importance of periodic reporting: We have previously stated the terminal should report to the PBX the current quality conditions it is observing. Loss and jitter metrics are sent every 0.5 sec from the mobile terminal to the PBX. Link layer metrics are read at intervals of 0.125 sec, four times the frequency of the VoIP metrics. Since the link layer situation ultimately reflects in the quality seen at the application layer, we deemed it necessary to use higher resolution at this layer. The link layer metrics are averaged and sent with the network parameters in RTCP-like reports. The timings are a tradeoff between the measurement resolution and the CPU load on the PDA.

Signal strength: As we have seen the signal to noise ratio is a good indicator of potential problems. Therefore given a dependable reading, we only need to record its value and scale it to our handover score. Unfortunately the signal strength reading from the PDAs tends to bottom out long before we loose connectivity, and consequently only makes a small contribution to the handover score, which is a limitation of the terminals we used. A positive signal strength is simply added to the score, in our experiments with the HP terminal this varied between +90 and 0.

Loss: We have seen from our off-line PESQ experiments that eight losses are sufficient to reduce the quality from "excellent" to "good/fair". A 20ms packetisation represents 50 packets per second, therefore a loss of eight packets corresponds to a loss percentage of 16% percent. In each second there are two reports (0.5 sec per report), therefore a loss of 8% should be taken into account. A score of -10 is attributed to this degree of loss for each interval and an additional -10 is added if this level spans over two intervals.

Jitter: We have seen increasing jitter was the best indicator we had of poor upcoming quality. In an open system it is easy to calculate the mean and variance of the VoIP stream by observing packet interarrival times. However, in our full system jitter estimates are returned from a commercial VoIP encoding and playout system called the GIPS Engine[1]. We were uncertain about the exact units returned, but found from experimentation that, values between 0-68 signified good conditions, whilst those between 69-80 were interpreted as neutral, 81-93 as bad and over 94 as poor. To find these values we loaded the network as described in the emulated cases, and observed the values reported. We attributed scores of +10 to the good conditions (i.e. a positive score), 0 to the neutral situation, -10 and -20 for the poor and very poor situations respectively. Similarly if these conditions span over two intervals, this is accounted for in the score.

[1] http://www.globalipsound.com

RTCP losses: It is important that the PBX has information about the state of the mobile terminal, as if the PBX is not receiving reports then the mobile is probably having reception problems and as we have seen, more likely worse than those seen at the PBX. Therefore sending regular reports from the mobile terminal to the PBX probes the 802.11 quality, and reports indicate potential problems. We chose three or more consecutive losses as sufficiently significant to reduce the score. Two or more report losses are interpreted as poor conditions between the handset and PBX and a score of -10 is attributed to this condition.

Transmission rates: As the system reduces the rate we would ideally like to reflect this in the handover score. In particular changes to the lower rates i.e. 2 and 1 Mbits/sec should reduce the score as the probability of a connection loss increases. However the PDA terminals did not reliably report this value to our application, hence we could not include it into our score function. As we have shown, laptops in the testbed setup gave IEEE transmission rates that we could have been used.

Handover score weighting: Since we have chosen to use a linear combination of the metrics, it is simple a matter of combining the above metrics into a single score value.

$$Handover\ score = Signal + Loss + Jitter + Report\ losses$$

Handover score values: For convenience our implementation uses a handover score that varies between -100 and 100. A large positive value indicates good quality. The user enters a threshold value and a handover will occur when the score falls below this level. We chose +30 as a default from experimental testing found it to be satisfactory. By increasing the threshold, average quality will improve but at greater expense since the system will hand over the call to the GSM system earlier. Conversely by decreasing the threshold, GSM expenses will be reduced but the periods of degraded audio quality will be longer. It was necessary to smooth these scores in some cases by considering two intervals, however we attribute this to using some combinations of hardware. This was not necessary in the emulated testbed setup.

5 System evaluation

In this section we describe the procedure we used to evaluate the performance of the handover trigger in a real setting. Figure 11 shows the reports are combined and sent to the PBX. Figure 12 shows the target system into which we have integrated our handover trigger module. The server and terminal are from Optimobile[2] and comprises a system capable of voice roaming. The PBX connects to the local Ethernet *and* to the PSTN providing connectivity to the GSM network. We used an HP-6340 PDA terminal with 802.11 and GSM interfaces. Multi-path probing, by sending data over both interfaces simultaneously is not performed in this setup.

[2] http://www.optimobile.se

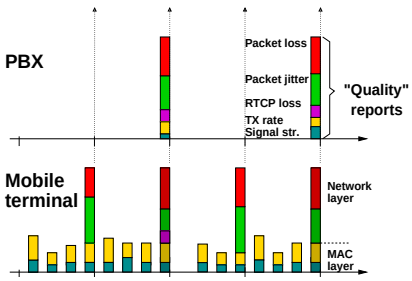

Fig. 11. Quality reports are sent periodically from the mobile to the PBX.

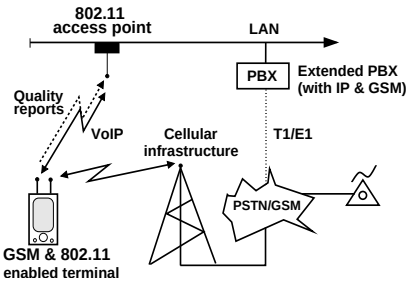

Fig. 12. The complete system used. Our module resides in the terminal & PBX.

When evaluating the trigger performance, we need to match our objective score with the listening judgment of a test subject. The role of the test subject is to indicate at what time the 802.11 quality becomes unacceptable. Therefore, we called from the mobile terminal over the 802.11b network using VoIP via PBX to the public PSTN to a phone picking up constant speech. Using the 802.11b network, the test subject walked out of the office waiting for a handover to occur, the walk is shown in Figure 6. The handover was never performed, rather when the score fell below the chosen threshold the time was recorded in a file. Later we compared the trigger time with the time when the test subject indicated unacceptable quality. Ideally, the trigger time should precede the subjective time by five seconds since it requires approximately this time to establish a PSTN connection. Note that it is possible to subjectively judge whether the handover occurred too late i.e. perceived poor quality, however not too early unless one examines the recorded times.

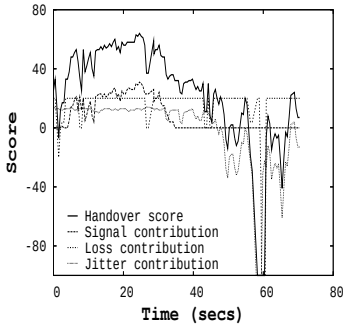

Fig. 13. Handover score when walking out of the office. The bold line is the score, the other lines its contributors.

Perceived quality started good	Timely HO	Late HO
and became bad	**68**	10
Perceived quality started good	Unnecessary HO	No HO
and remained good	7	**15**

Fig. 14. 100 trial handover (HO) results showing 83% success. The bold values show optimal decisions.

Figure 13 shows the result of one coverage experiment whilst Figure 14 shows the results of 100 experiments. In most cases the quality did not deteriorate at the same physical location, due to radio interference and imperfect terminal software. In 68 cases the trigger released on time as desired. In 10 cases the trigger came too late, i.e. the subject perceived poor quality for a brief period while waiting for handover to occur. In 7 cases the trigger suggested an unnecessary handover, i.e. the call became more expensive than necessary. The remaining 15 runs never triggered handover which is optimal. Therefore in 83% of the cases the algorithm made the ideal decision. In 10% of the cases quality temporarily deteriorated because the handover came late, this is inconvenient but not fatal.

6 Related work

Calvagna et al. present an overview of handover issues with a focus on hybrid mobile data networks [10]. They propose a neural network solution for handovers to/from 802.11 networks to GPRS networks and show its performance to be good. The E-Model as standardized by the ITU-T allows for the prediction of voice quality based on network QoS parameters [4]. However, it is not useful for our purposes because it does not take the signal strength and delay jitter into account. Very recent work by Hoene et al. propose a real-time implementation of PESQ called PESQlite [3]. It reduces the complexity by making simplifications to the PESQ algorithm e.g. using constant length test samples and non time alignment of the degraded samples. Our off-line method has a slightly different purpose, it is to obtain a mapping between consecutive packet loss and the PESQ MOS score. Dimitriou et al. state that interference and users moving out of range as limiting factors for good VoIP quality in WLANs [1]. Their solution is to use better speech coding and suggest an enhanced version of G.711 to make the speech more resilient to loss. Kashihara and Oie developed a WLAN handover scheme for VoIP that makes use of MAC-layer information on the number of retransmissions of the voice packets [11]. If this number exceeds a certain threshold, the system switches to multi-path transmission of the packets. As soon as one of the WLAN interfaces reaches a stable condition, it can be used for single-path transmission. In Fitzpatrick et al. propose a transport layer handover mechanism using the stream control transmission protocol (SCTP) [7]. The mechanism uses the multi-homing feature of SCTP and measures the network performance metrics by sending probes. Handover decisions are based on speech quality estimations utilizing the ITU-T's E-Model.

7 Conclusions, future work and acknowledgments

The goal of this work was to map measurable parameters to speech quality in order to implement triggers for voice handovers. The solution was integrated into an existing system for evaluation. We have shown that automatic network roaming worked ideally in 83% of the trials we conducted. The results of the experiments can be changed by choosing the threshold value of the trigger. More

precisely the balance between remaining in the 802.11 network longer and switching earlier can be chosen. Therefore the threshold value can be seen as a monetary selection. The fraction of expensive calls may be reduced by lowering the threshold but this will increase the periods of deteriorated quality. In the case where the mobile roams from the cellular to the 802.11 network, i.e. enters a LAN. A different approach is needed where probing the quality before handing over would be more appropriate. This work has been partly supported by the European Union under the E-Next Project FP6-506869, the Vinnova SIBED program in Sweden and the Austrian government's Kplus competence center program. We are very grateful to Optimobile AB for allowing us to use their system in the testing and evaluation phases. Thanks to Bengt Ahlgren, Pekka Hedqvist, Henrik Lundqvist, Per Gunningberg, Gunnar Karlsson, Martín Varela and Thiemo Voigt for their valuable comments on this paper.

References

1. E. Dimitriou and P. Sörqvist. Internet Telephony over WLANS. Technical report, Global IP Sound, Sept. 2003. http://www.globalipsound.com/solutions/wlan.usta_paper.pdf.
2. F. Hammer, P. Reichl, and T. Ziegler. Where Packet Traces meet Speech Samples: an Instrumental Approach to Perceptual QoS Evaluation of VoIP. In *IEEE International Workshop on Quality of Service IWQOS 2004*, pages 273–280, Montreal, Canada, June 2004.
3. C. Hoene. *Internet Telephony over Wireless Links.* PhD thesis, Technical University of Berlin, Germany, Dec. 2005.
4. International Telecommunication Union. The E-model, a computational model for use in transmission planning. Recommendation G.107, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Dec. 1998.
5. International Telecommunication Union. Appendix I: A high quality low-complexity algorithm for packet loss concealment with G.711. *ITU-T Recommendation G.711, Appendix I*, Sept. 1999.
6. International Telecommunication Union. Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. Technical report, Telecommunication Standardization Sector of ITU, Feb. 2001.
7. John Fitzpatrick and Sen Murphy and John Murphy. An Approach to Transport Layer Handover of VoIP over WLAN. In *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, USA, Jan. 2006.
8. A. Kamerman and L. Monteban. WaveLAN-II: A High-performance wireless LAN for the unlicensed band. *Bell Lab Technical Journal*, pages 123–140, Apr 1990.
9. M. Lacage, M. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, Venice, Italy, Oct. 2004.
10. K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Y. J. Mäkelä, R. P. R., and J. V. J. Handoff in hybrid mobile data networks. *IEEE Personal Communications Magazine*, pages 34–47, Apr. 2000.
11. Shigeru Kashihara and Yuji Oie. Handover Management based upon the number of retries for VoIP in WLANs. In *Proc. IEEE Vehicular Technology Conference (VTC2005)*, May 2005.

Paper I

Martín Varela, Ian Marsh and Björn Grönvall.
A Systematic Study of PESQ's Performance from a Networking Perspective.
In *Proceedings of Measurement of Speech and Audio Quality in Networks (MESAQIN)*, Prague, Czech Republic, May 2006.

"I find myself rearranging my points of view
There isn't much I could do
Despite my fear it helps to
Share my nostalgia with you
Tomorrow I remember yesterday
Tomorrow, remember yesterday"

Nostalgia - The Chameleons

A systematic study of PESQ's behavior (from a networking perspective)

Martín Varela[1], Ian Marsh[1], and Björn Grönvall[1]
mvarela@sics.se, ianm@sics.se, bg@sics.se

Swedish Institute of Computer Science (SICS) Kista, Sweden

Abstract. In this paper we study, in a systematic way, how the behavior of PESQ estimations vary with the network loss process. We assess the variability of these estimations with respect to the network conditions and the speech content. We judge the estimation accuracy with subjective tests and the ITU's single-sided measure.

1 Introduction

PESQ [ITU01], the ITU-T's Perceptual Evaluation of Speech Quality is among the most widely used objective voice assessment tools in telecommunications and IP networks. Several commercial offerings incorporate it as a central component for voice over IP quality assessment. In terms of accuracy, i.e. correlation with subjective assessments, it has an advantage over other purely objective quality metrics [Psy01]. While it does perform very well for traditional telephony applications, it has been noted that its performance decreases when used on VoIP scenarios, which exhibit bursty losses [Pen02,Psy01].

In this paper we take a systematic, black-box approach to analyzing the performance of PESQ, from a networking perspective. We focus on the impact of the packet loss process. However, as far as the voice quality itself is concerned, we consider that the dominant degradation factor will be the network losses. For our experiments, we considered G.711 streams with and without packet loss concealment (PLC). To this end, we have created a basic testing framework which helps prepare and carry out tests, both objective and subjective. Our goals within this work is assessing the performance of PESQ in two different VoIP settings, namely wired and wireless networks.

We have studied the performance of PESQ under a variety of both uniform and bursty losses. For the latter case, we have also conducted subjective assessments in order to derive an idea of PESQ's performance in relation to real user tests. The general idea is pre-generate loss sequences, for various distributions and examine the scores given by PESQ. This treats the processing of PESQ as a black box as explained. We have additionally studied how PESQ's results compare to those obtained with the ITU's P.563 single-sided metric [ITU04]. The rest of the paper is organized as follows. Section 2 presents a description of the experiments we carried out. The results we obtained are discussed in Section 3. Finally, we conclude the paper and discuss future work in Section 4.

2 Description of the experiments

As mentioned above, we have focused our experiments on the behavior of PESQ under different loss processes that can be found on wired and wireless Internet connections. We used G.711 coding, both with and without PLC. The experiments we conducted can be classified, according to the scenarios considered, as follows.

1. Uniform losses
2. Gilbert losses, large loss space
3. Gilbert losses, restricted loss space

2.1 Uniform losses

The first loss model we used for our study is that of uniform loss distribution. While this is a very simplistic model, since it assumes no temporal correlation between consecutive losses, it can be used to model network behavior when the loss rate is relatively low [HW99,MCA01]. We performed several tests using uniform loss sequences. The first was to see the PESQ scores as the loss rate was increased. We assessed ten different samples, each with ten different loss sequences for each loss rate considered. We then calculated the average of the 100 PESQ scores obtained, as well as their variance. The uniform loss model was also used to study the variations of PESQ scores observed when a given loss sequence occurs in different positions within a voice segment. Essentially, this means what is the difference in PESQ scores when certain parts of speech are lost due to packet loss.

2.2 Gilbert losses, large loss space

The second loss model used was a simplified version of the Gilbert model [Gil60]. This simplified version is widely used in the literature [SCK00,BFPT99], since it provides an accurate, yet relative easy method of generating bursty loss sequences.

The Gilbert model: In this model the channel has two states shown in Figure 1), one in which the transmission is successful and another in which errors occur. The states 0 and 1 represent a packet arrival and loss respectively. We denote by p the probability of a packet being lost given that the previous one arrived. The probability $1 - q$ is that of losing a packet given that the previous one was also lost.

The relationship between the parameters in the model and the ones we use in this paper, the loss rate (LR) and the mean loss burst size (MLBS) is as follows:

$$p = \frac{1}{\text{MLBS}} \frac{\text{LR}}{1 - \text{LR}}, \qquad q = \frac{1}{\text{MLBS}}. \tag{1}$$

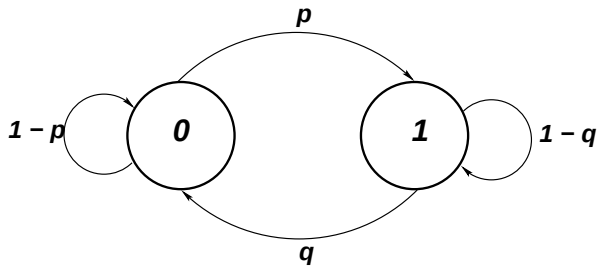

Fig. 1. The simplified Gilbert model. When in state 0, the transmission is error-free. In state 1, a loss occurred. Note that the transition from state 0 to state 1 implies a loss, and that in the opposite direction, it implies that the packet arrived.

Note that if there are losses (at least one) and if not every transmission is a loss, then MLBS > 1 and $0 <$ LR < 1, leading to $0 < p$ and $q < 1$.

PESQ standard samples are 8 seconds long. At 20 ms packetisation this corresponds to 400 packets. One problem we found when using the Gilbert model was to generate loss sequences for such a low number of packets. the two state process needs more than 400 packets to reach a steady state distribution. Although we could have generated longer sequences, we decided to keep to the 400 sequence length. This generally induces a difference between the target values of LR and MLBS, and the actual values obtained in the loss strings. This, in turn, adds some variance to the tests. We dealt with this issue when working on the restricted loss space described below.

The experiments: We considered a very large loss space, with loss rates ranging from 0 to 50%, and with mean loss burst sizes ranging from 1 to 10 packets using 16 intermediate MLBS values. This loss space covers, and probably exceeds, most possible loss conditions that can be found for VoIP traffic. Considering all these combinations allowed us to consider loss sequences commonly found in both wired and wireless networks. In the latter, it is relatively common to experience very bursty losses, even for relatively low loss rates. One downside to using this space is that some of the combinations are not actually feasible when using 400-packet samples for the reasons stated above.

For each point of the loss space (816 in total), we generated 10 different sequences, and then processed 20 speech samples both with and without PLC. This gave us 400 degraded samples, for which we then calculated PESQ scores. This run implied 426000 PESQ executions, which needs about two seconds per execution. The total time for such an experiment was about 180 hours of computing time, using a Pentium IV with 1GB RAM as reference.

2.3 Gilbert losses, restricted loss space

As mentioned previously, using the Gilbert model presents some problems with the large loss space and with the (short) 400 packet samples. In order to improve the accuracy of our results, a possible solution would be to use longer speech samples, so that the Gilbert model implementation can converge to the target values. We performed tests to determine how long the samples should be in order for the loss model to converge. The results obtained indicate that between 3000 and 4000 packets would allow for good convergence. This, however, implies very long samples, which would exceed the sample length recommended for PESQ [ITU01]. Therefore, in order to use the standard 8-second samples and improve the accuracy of our measurements, the next sections will discuss:

- Remove infeasible LR and MLBS combinations
- Obtain more accurate 400 packet loss sequences

In order to eliminate the unfeasible loss conditions, we simply restricted the loss space, so that all LR and MLBS combinations would be feasible, see Figure 2. We also reduced the maximum loss rate and mean loss burst sizes to 30% and 6 packets respectively. As for the accuracy problem in the generated loss sequences, we needed to obtain several different sequences for each point in the loss space. In order to do this, we chose from a large pool of seeds for the random number generator, created sequences which were close enough for our purposes. We used a brute-force approach, however it would have been possible to generate such sequences with variation reduction techniques such as antithetic variables for example.

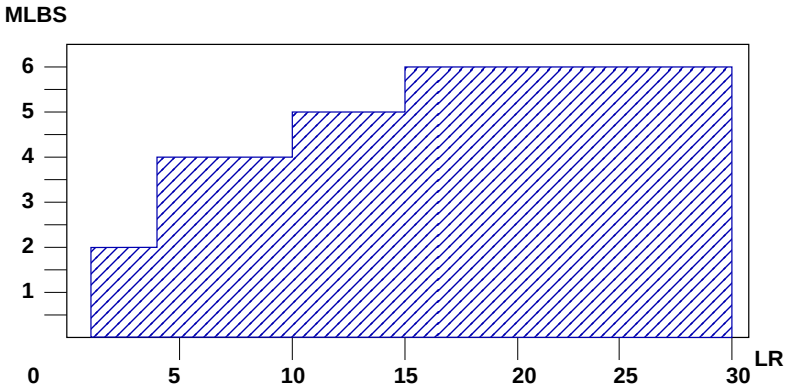

Fig. 2. The restricted loss space considered. Note that some combinations which are relatively common on wireless networks, like low loss rate and high burstiness, were removed to generate more accurate loss sequences.

We also run preliminary tests to determine whether variations in the speech samples induced more variability on PESQ results than variations on the loss sequence, or vice-versa. The results obtained indicate that the both parameters impose similar variance in the PESQ scores. We therefore used equal numbers of speech samples and loss sequences (15 each) for the experiments. Considering both PLC and non-PLC codings, we ended up 450 PESQ scores for each (LR, MLBS) point in the loss space.

2.4 Subjective assessment

In order to determine how the accuracy of PESQ's assessments varies with the network conditions, it is necessary to compare them with subjective assessments. We have, to this end, carried out an ITU P.800-based [ITU96] subjective assessment test. This test, while small in scale, provides a view of the relation between subjective scores and PESQ estimates over the loss space considered. We had 42 4-sample groups assessed, providing reasonable coverage of the restricted loss space. Of those 42 groups, 29 corresponded to samples without PLC, and the remaining 13 used PLC. Figure 3 shows the loss configurations considered during the test.

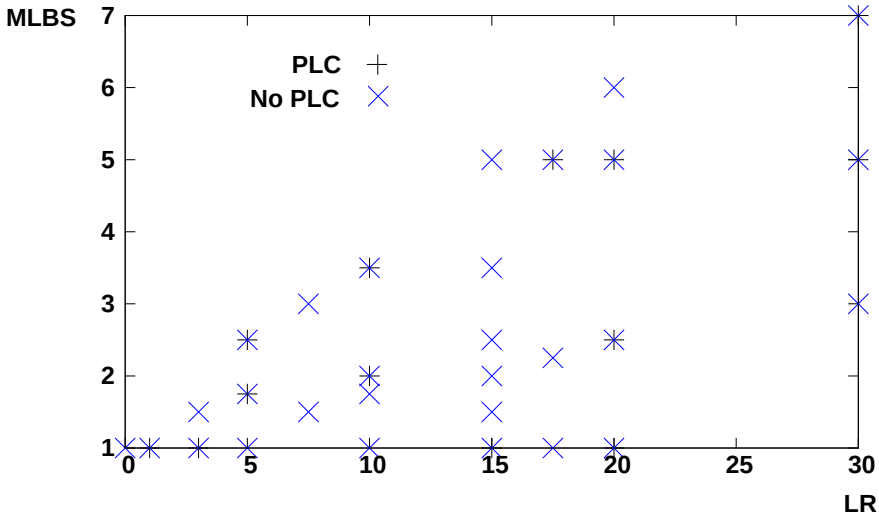

Fig. 3. Distribution of the loss conditions considered for the subjective tests.

We had 11 subjects assess the 168 speech samples, preceded by a series of warm-up samples, which included original-quality (i.e. non-degraded) samples. The samples and groups were randomly named and the groups were randomly sorted, so as to avoid any bias during the tests. The tests were driven automatically and the subjects wrote down their assessments on questionnaires. The

grading scale used was a 9-point one, and the results were later mapped into a 5-point scale for comparison with PESQ's output. Test times varied between about 30 and 45 minutes, and the test instructions suggested a mid-test rest of 5 to 10 minutes. The scores obtained were then statistically screened (e.g. for hearing problems) none of the subjects had to be discarded.

3 Experimental results

In this Section we summarise the main results obtained from the experimental descriptions above.

3.1 Results for the uniform loss scenarios

Using a uniform loss model gave us the data to analyse the PESQ in terms of loss rate only. The results obtained Figure 4 indicate that PESQ is over-estimating the perceived quality of the samples, especially for the higher loss rates. This was also observed later when analyzing the data from the subjective assessment tests and the results given by PESQ (see Section 3.4). These results can be improved by using PESQ-LQ [Rix03]. Seeing how much the variability in the results increases with the loss rate can help us decide under which conditions the use of PESQ is appropriate for a given telephony application.

We also studied the variation of PESQ scores as the same loss sequence was shifted in time with respect to the speech sample. We also studied the variance due to having different loss sequences with the same loss rate degrade a given sample. The maximum variations we found in these cases were in the order of 0.7 MOS points, which indicate that they should be noticeable by the average listener. Interestingly, this happened within just a 10-packet (200ms) shift in the loss sequence. Most of the time, however, the scores were very similar, irrespective of the changes to the loss sequence. Figure 5 shows how the PESQ scores vary for 10 different loss sequences applied to the same original sample.

3.2 Results for the large Gilbert loss space

In this section we discuss the main findings from the experiments run on the large Gilbert loss space. Figures 6 and 7 show the median PESQ scores calculated over the whole loss space, with and without PLC respectively. We can observe how the quality falls, as expected, with both increasing LR and MLBS. Also, it is clear that the while the LR is the dominant parameter, a bursty loss process can seriously impair the quality from a PESQ perspective. Naturally, the use of PLC allows for a smoother quality degradation for both loss types which is especially noticeable at low loss rates. In the non-PLC case, the drop in quality over the first 10 to 20% LR is noticeably more steep than when PLC is used.

Also interesting is the fact that the quality decreases more steeply when the LR values are low, and then the degradation becomes less pronounced. It also appears that the curve for the median PESQ as a function of LR (for fixed MLBS

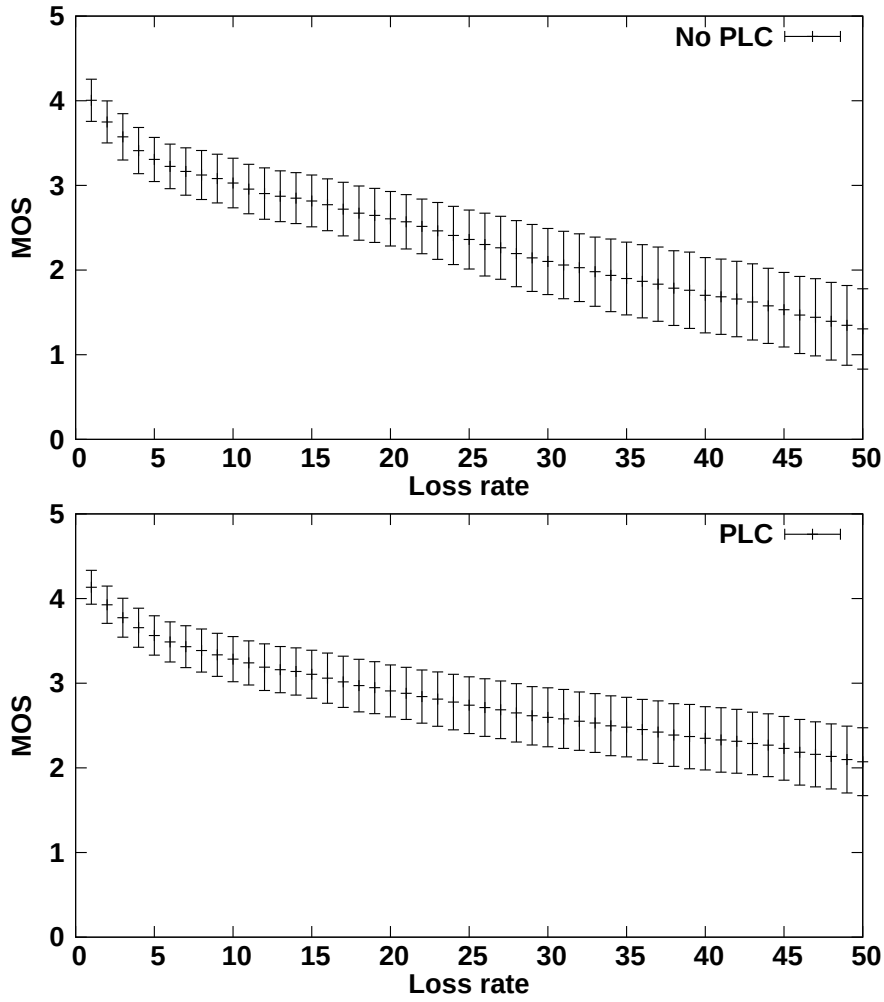

Fig. 4. PESQ scores as a function of the loss rate using a uniform loss model. Note that the estimates remain high even for very high loss rates. Also, the variability in the estimates is slightly higher when PLC is not used, although in both cases is relatively small.

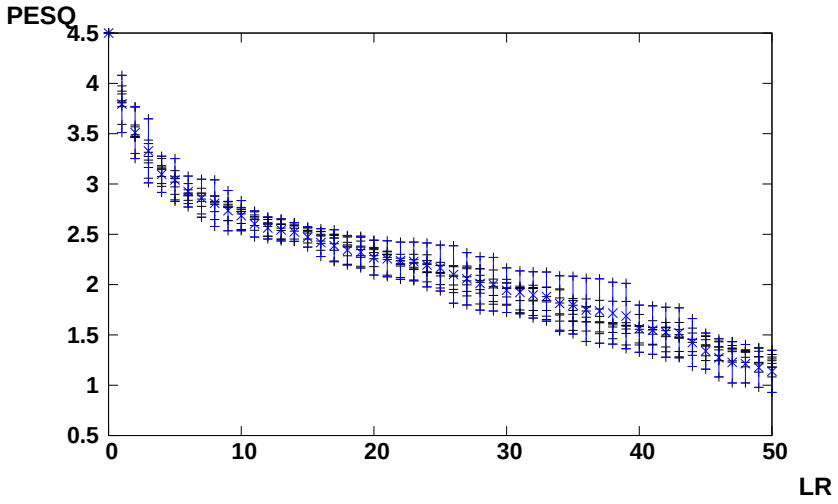

Fig. 5. Example of the variation of PESQ scores for 10 different loss sequences applied to the same original sample. Note that the variations is generally small, however a maximum variations of about 0.7 MOS points can be observed.

values) is composed of two roughly linear segments, as can be seen in Figures 9 and 10.

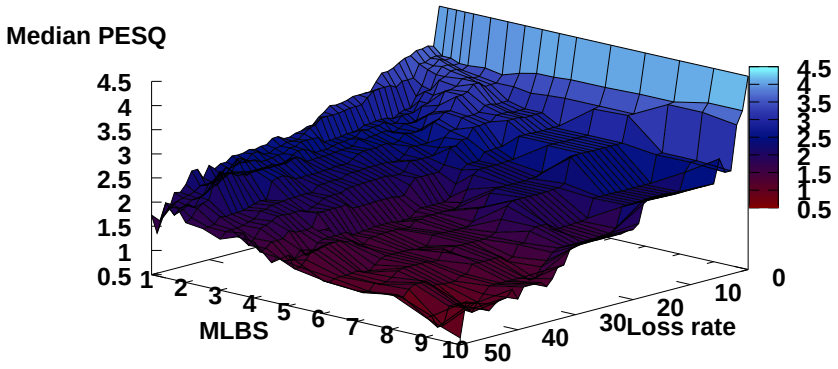

Fig. 6. Median PESQ scores over the complete loss space considered, with PLC. The median was calculated over 200 PESQ scores for each (LR,MLBS) combination.

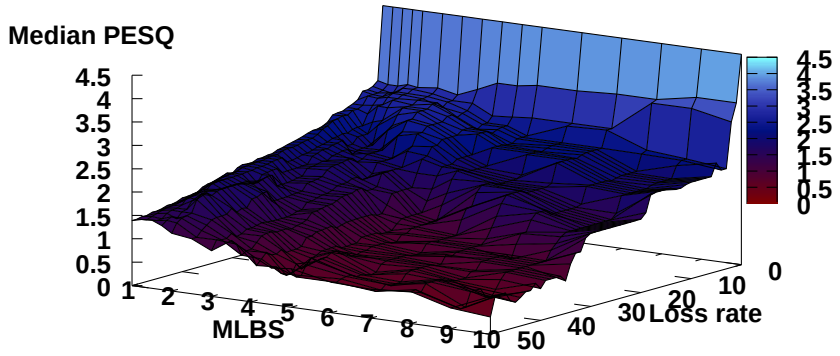

Fig. 7. Median PESQ scores over the complete loss space considered, without PLC. The median was calculated as in the PLC case. Note the steeper descent of the quality as the loss rate increases when no PLC is used.

3.3 Results for the restricted Gilbert loss space

As mentioned in Section 2.3, covering the whole loss space implies a certain decrease in the accuracy of the results obtained. To remedy this, we have studied a more restricted loss space, and increased the accuracy of the Gilbert model's output. The results obtained present a more accurate view of PESQ's behavior as the network conditions change. An interesting first result, is that the overall variability in the estimations is significantly reduced.

In Figure 8 we can compare the absolute deviations of the estimations over both the large and the restricted loss space. The accuracy of the estimations is much more even for the latter case, as above especially when network conditions degrade.

Figures 9 and 10 show plots of the median PESQ scores as a function of LR, with the absolute deviation also plotted. Interestingly, it would seem that not using PLC induces a greater variation into the results. We still do not know the reason for this. However, the absolute deviation is small in most cases. This hints that the median can be a relatively good approximation for the PESQ scores of the 225 samples considered for each point. We've also calculated interquartile ranges, and also found them to be small.

3.4 Comparison with subjective scores

Although the subjective campaign we carried out was relatively small, it does provide some insight into the actual accuracy of the PESQ assessments as the loss conditions vary. Figure 11 shows the MOS value obtained for each sample, along with their standard deviations.

The overall correlation of PESQ and subjective scores was 0.867, which is a similar value to the one reported in [Psy01]. The scatter plot in Figure 12 suggests that the performance of PESQ, in terms of correlation with subjective scores,

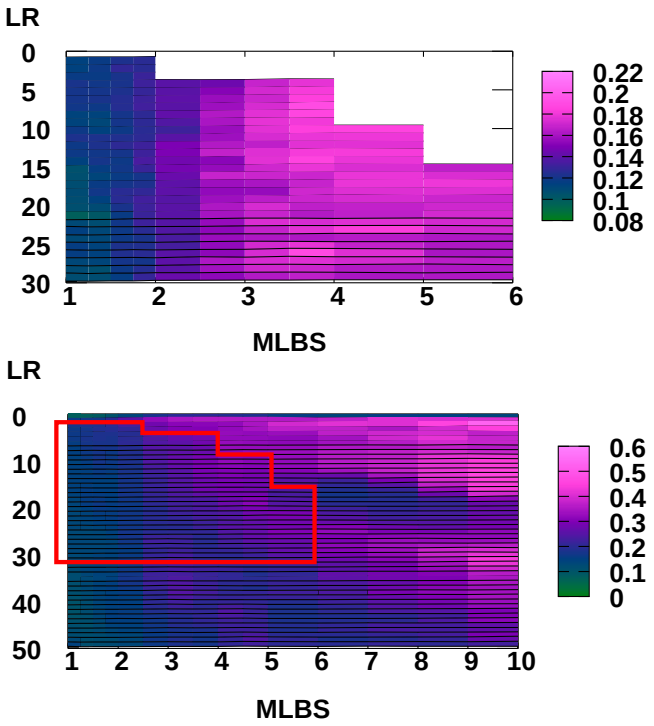

Fig. 8. Absolute deviation of PESQ scores at each point of the loss space. The red outline in the large space indicates the restricted space. Note how the variability of the results has decreased.

remains relatively stable even when the network conditions degrade. Correlation coefficients for each subset were of 0.751 and 0.733, respectively.

When comparing the actual estimates, it is easy to see that, even as the correlation remains relatively high, there are variations in its behavior with respect to the subjective scores. In Figure 13 we can see that PESQ is over-estimating the quality when the losses are small. As the losses become more bursty, PESQ's estimations drop faster than the actual MOS, so therefore PESQ underestimates for the highly bursty losses. The best estimations correspond to moderately bursty losses.

3.5 An informal performance comparison of PESQ and P.563

We performed a short comparison on the performances of PESQ and the P.563 single-sided assessment technique, in order to obtain an idea of how reasonable the P.563 estimations were. We believe that, although both metrics work under different conditions, access to the reference signal should provide better estimates of the quality. The P.563 estimations of the degraded samples used

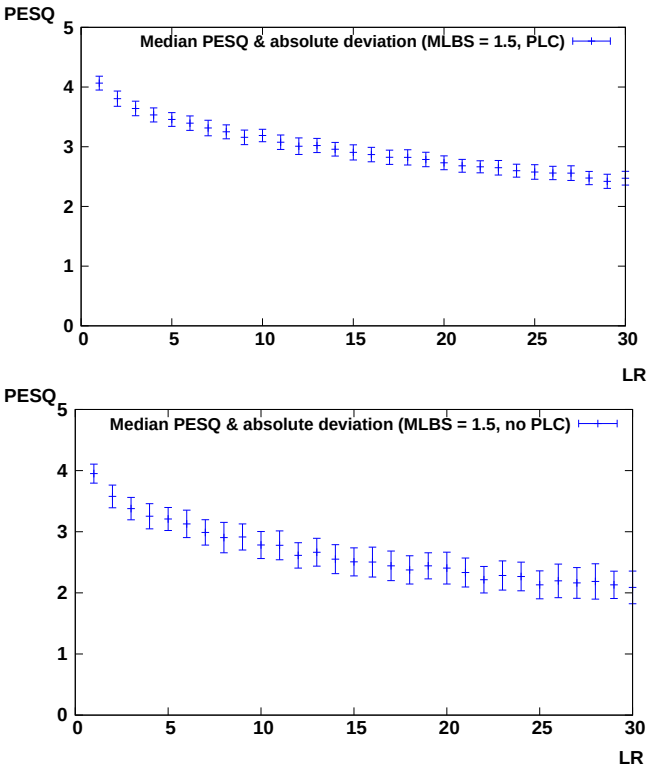

Fig. 9. Median PESQ scores and absolute deviation as a function of loss rate. MLBS = 1.5 packets, and both PLC and non PLC cases are shown.

for the subjective tests presented quite different results to that of PESQ. The single-sided metric underestimated the quality under low losses, and gradually approached the MOS values as the loss rate and burstiness increased (slightly overestimating for very bursty losses). This can be seen in Figure 14.

In terms of correlation with the subjective scores, P.563 did not provide results as good as PESQ's. The overall correlation was 0.795. While not insignificant it is worth while to use PESQ where possible.

4 Conclusions and future work

In this paper we have presented a systematic study of the behavior of PESQ as the network loss conditions vary. The main goals of this study are to gain a better understanding of the circumstances under which PESQ is able to provide accurate assessments, and to also understand what kind of adjustments need to be made when the accuracy degrades.

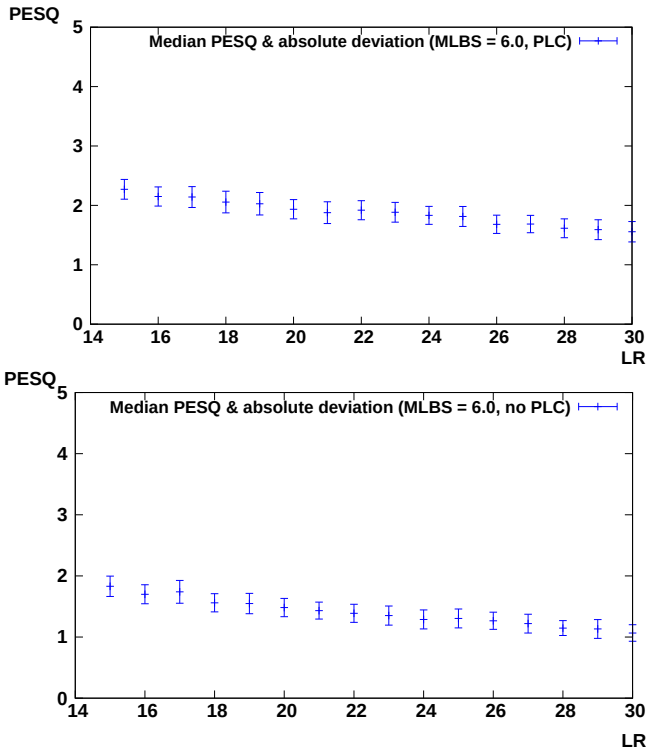

Fig. 10. Median PESQ scores and absolute deviation as a function of loss rate. MLBS = 6.0 packets, and both PLC and non PLC cases are shown.

We have analyzed the variability of PESQ scores under several different conditions, and found it to be relatively small, which opens the door for performing PESQ-like, single-sided estimations of the quality of a voice stream. We've also analysed the accuracy of PESQ as the network conditions change, by means of comparison with subjective scores. In particular, it seems that PESQ maintains reasonable correlation with subjective scores even when the network conditions are poor. Also, the deviations it exhibits from the subjective scores seem systematic, which suggests that a simple compensation factor might be found (for instance, derived from the network conditions) and used to further improve the results.

An informal performance comparison has been performed between PESQ and the P.563 single-sided metric, and with the data available, the results indicate that PESQ provides more accurate quality estimates. As stated this seems natural if the signal processing has access to the original samples.

As for possible research directions in this area, we consider that more subjective assessments similar to the ones presented here would greatly improve our understanding of PESQ, and probably allow for improvements to be made, as

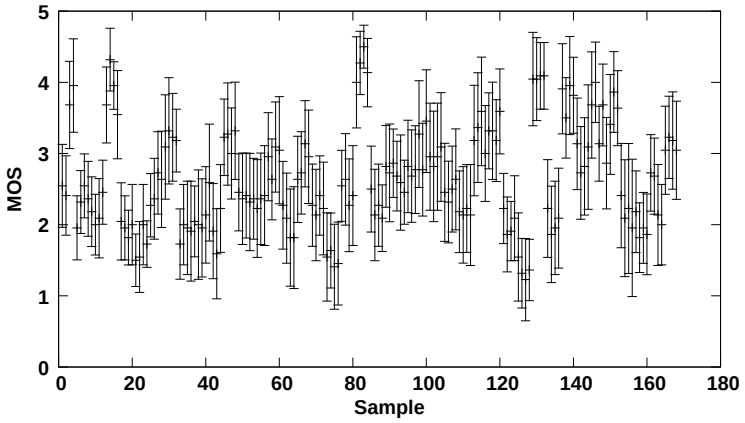

Fig. 11. MOS values and their respective standard deviations for all the samples tested.

mentioned above. We are also working on the development of loss-based single-sided metric based on PESQ, to be used in real-time environments.

References

[BFPT99] J-C. Bolot, S. Fosse-Parisis, and D.F. Towsley. Adaptive FEC–Based Error Control for Internet Telephony. In *Proceedings of INFOCOM '99*, pages 1453–1460, New York, NY, USA, March 1999.

[Gil60] E. Gilbert. Capacity of a Burst–loss Channel. *Bell Systems Technical Journal*, 5(39), September 1960.

[HW99] D. Hands and M. Wilkins. A Study of the Impact of Network Loss and Burst Size on Video Streaming Quality and Acceptability. In *Interactive Distributed Multimedia Systems and Telecommunication Services Workshop*, October 1999.

[ITU96] ITU-T Recommendation P.800. Methods for Subjective Determination of Transmission Quality, August 1996.

[ITU01] ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (Pesq), an Objective Method for End-To-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs, 2001.

[ITU04] ITU-T Recommendation P.563. Single–ended Method for Objective Speech Quality Assessment in Narrow–band Telephony Applications, May 2004.

[MCA01] S. Mohamed, F. Cervantes, and H. Afifi. Integrating Networks Measurements and Speech Quality Subjective Scores for Control Purposes. In *Proceedings of IEEE INFOCOM'01*, pages 641–649, Anchorage, AK, USA, April 2001.

[Pen02] S. Pennock. Accuracy of the perceptual evaluation of speech quality (PESQ) algorithm. In *Measurement of Speech and Audio Quality in Networks Line Workshop, MESAQIN '02*, January 2002.

[Psy01] Psytechnics Ltd. PESQ: an Introduction. http://www.psytechnics.com, September 2001.

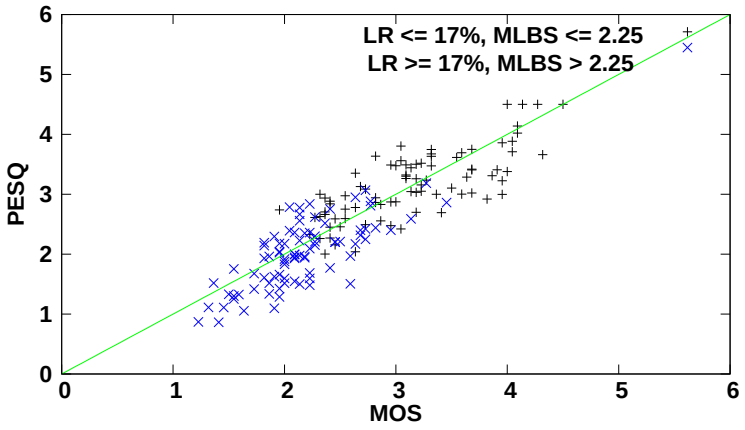

Fig. 12. PESQ scores vs MOS values.

[Rix03] Antony W. Rix. Comparison between subjective listening quality and P.862 PESQ score. In *Proc. Measurement of Speech and Audio Quality in Networks (MESAQIN'03)*, Prague, Czech Republic, May 2003.

[SCK00] H. Sanneck, G. Carle, and R. Koodli. A Framework Model for Packet Loss Metrics Based on Loss Runlengths. In *Proceedings of the SPIA/ACM SIGMM Multimedia Computing and Networking Conference*, pages 177–187, San Jose, CA, January 2000.

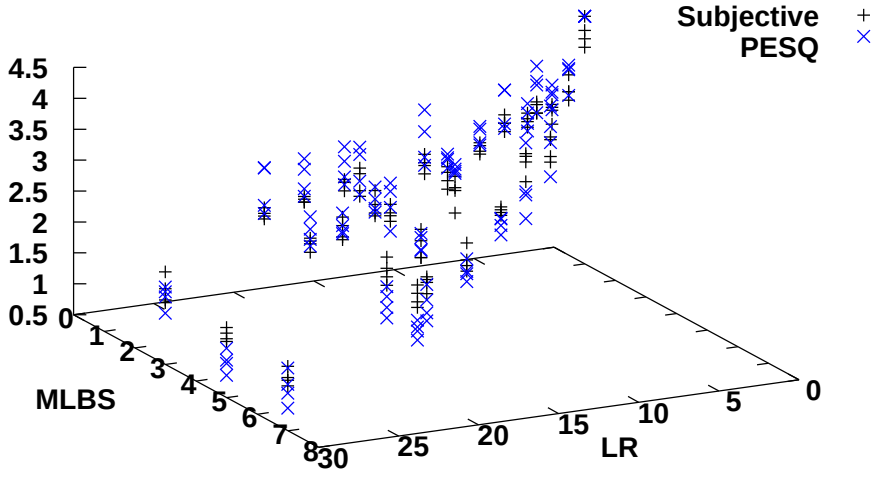

Fig. 13. PESQ scores and MOS as a function of the loss rate and the mean loss burst size. We can see that PESQ overestimates the quality when the burstiness is low, and underestimates it when the losses are bursty. The best estimations are those corresponding to moderately bursty losses.

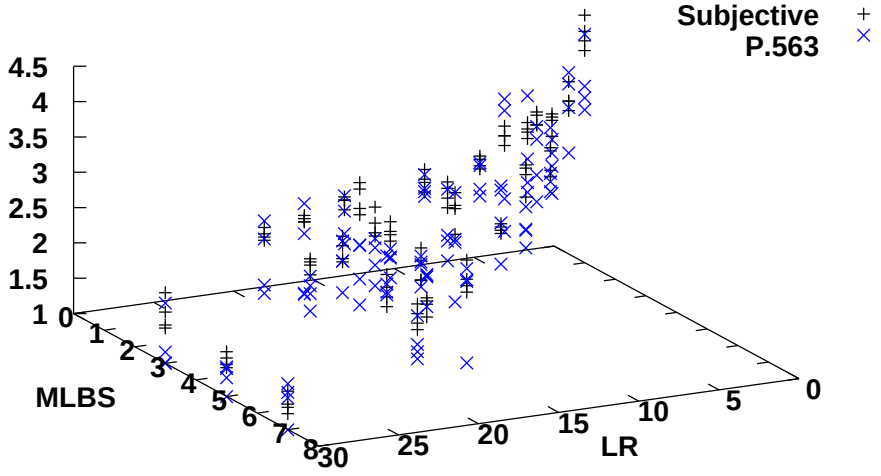

Fig. 14. P.563 scores and MOS as a function of the loss rate and the mean loss burst size.

Swedish Institute of Computer Science

SICS Dissertation Series

1. Bogumil Hausman, Pruning and Speculative Work in OR-Parallel PROLOG, 1990.

2. Mats Carlsson, Design and Implementation of an OR-Parallel Prolog Engine, 1990.

3. Nabiel A. Elshiewy, Robust Coordinated Reactive Computing in SANDRA, 1990.

4. Dan Sahlin, An Automatic Partial Evaluator for Full Prolog, 1991.

5. Hans A. Hansson, Time and Probability in Formal Design of Distributed Systems, 1991.

6. Peter Sjödin, From LOTOS Specifications to Distributed Implementations, 1991.

7. Roland Karlsson, A High Performance OR-parallel Prolog System, 1992.

8. Erik Hagersten, Toward Scalable Cache Only Memory Architectures, 1992.

9. Lars-Henrik Eriksson, Finitary Partial Inductive Definitions and General Logic, 1993.

10. Mats Björkman, Architectures for High Performance Communication, 1993.

11. Stephen Pink, Measurement, Implementation, and Optimization of Internet Protocols, 1993.

12. Martin Aronsson, GCLA. The Design, Use, and Implementation of a Program Development System, 1993.

13. Christer Samuelsson, Fast Natural-Language Parsing Using Explanation-Based Learning, 1994.

14. Sverker Jansson, AKL - A Multiparadigm Programming Language, 1994.

15. Fredrik Orava, On the Formal Analysis of Telecommunication Protocols, 1994.

16. Torbjörn Keisu, Tree Constraints, 1994.

17. Olof Hagsand, Computer and Communication Support for Interactive Distributed Applications, 1995.

18. Björn Carlsson, Compiling and Executing Finite Domain Constraints, 1995.

19. Per Kreuger, Computational Issues in Calculi of Partial Inductive Definitions, 1995.

20. Annika Waern, Recognising Human Plans: Issues for Plan Recognition in Human-Computer Interaction, 1996.

21. Björn Gambäck, Processing Swedish Sentences: A Unification- Based Grammar and Some Applications, 1997.

22. Klas Orsvärn, Knowledge Modelling with Libraries of Task Decomposition Methods, 1996.

23. Kia Höök, A Glass Box Approach to Adaptive Hypermedia, 1996.

24. Bengt Ahlgren, Improving Computer Communication Performance by Reducing Memory Bandwidth Consumption, 1997.

25. Johan Montelius, Exploiting Fine-grain Parallelism in Concurrent Constraint Languages, 1997.

26. Jussi Karlgren, Stylistic experiments in information retrieval, 2000.

27. Ashley Saulsbury, Attacking Latency Bottlenecks in Distributed Shared Memory Systems, 1999.

28. Kristian Simsarian, Toward Human Robot Collaboration, 2000.

29. Lars-åke Fredlund, A Framework for Reasoning about Erlang Code, 2001.

30. Thiemo Voigt, Architectures for Service Differentiation in Overloaded Internet Servers, 2002.

31. Fredrik Espinoza, Individual Service Provisioning, 2003.

32. Lars Rasmusson, Network capacity sharing with QoS as a financial derivative pricing problem: algorithms and network design, 2002.

33. Martin Svensson, Defining, Designing and Evaluating Social Navigation, 2003.

34. Joe Armstrong, Making reliable distributed systems in the presence of software errors, 2003.

35. Emmanuel Frecon, DIVE on the Internet, 2004.

36. Rickard Cöster, Algorithms and Representations for Personalised Information Access, 2005.

37. Per Brand, The Design Philosophy of Distributed Programming Systems: the Mozart Experience, 2005.

38. Sameh El-Ansary, Designs and Analyses in Structured Peer-to-Peer Systems, 2005.

39. Erik Klintskog, Generic Distribution Support for Programming Systems, 2005.

40. Markus Bylund, A Design Rationale for Pervasive Computing - User Experience, Contextual Change, and Technical Requirements, 2005.

41. Åsa Rudström, Co-Construction of hybrid spaces, 2005.

42. Babak Sadighi Firozabadi, Decentralised Privilege Management for Access Control, 2005.

43. Marie Sjölinder, Age-related Cognitive Decline and Navigation in Electronic Environments, 2006.

44. Magnus Sahlgren, The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces, 2006.

45. Ali Ghodsi, Distributed k-ary System: Algorithms for Distributed Hash Tables, 2006.

46. Stina Nylander, Design and Implementation of Multi-Device Services, 2007.

47. Adam Dunkels, Programming Memory-Constrained Networked Embedded Systems, 2007.

48. Jarmo Laaksolahti, Plot, Spectacle, and Experience: Contributions to the Design and Evaluation of Interactive Storytelling, 2008.

49. Daniel Gillblad, On Practical Machine Learning and Data Analysis, 2008.

50. Fredrik Olsson, Bootstrapping Named Entity Annotation by Means of Active Machine Learning: a Method for Creating Corpora, 2008.