QoS-based Playout Control in IP Telephony over NGNs

Ing. Fabrizio BOI

Advisor: Dott. Ing. Luigi ATZORI
Curriculum: ING-INF/03 - Ph.D. in Electronic and Computer Engineering

XXII Cycle
March 2010
To my girlfriend and
our future.
# Contents

**Introduction** ................................................................................................................................................. 5

Related Papers ................................................................................................................................................. 10

**Chapter 1 – Voice over Internet Protocol** ................................................................................................. 11

1.1 Transport Layer: the use of UDP ........................................................................................................... 11

1.2 Real time Transport Protocol (RTP) ..................................................................................................... 12

1.3 Codecs ......................................................................................................................................................... 15

1.3.1 PCM Codec: G.711 ........................................................................................................................... 16

1.3.2 ADPCM Codec: G.726 ....................................................................................................................... 16

1.3.3 CELP Codec: G.729 ........................................................................................................................... 17

1.3.4 Voice Activity Detection (VAD) ....................................................................................................... 18

1.4 Signaling Protocol ....................................................................................................................................... 19

1.4.1 H.323: a brief description ................................................................................................................. 20

1.4.2 SIP: Session Initiation Protocol ....................................................................................................... 21

1.4.2.1 SIP Session Call ............................................................................................................................ 24

1.4.2.2 SIP Messaging Syntax: an overview ............................................................................................ 26

1.4.2.3 SIP Interworking .......................................................................................................................... 29

**Chapter 2 – QoS in VoIP services** .............................................................................................................. 32

2.1 Key Performance Indicators for QoS ...................................................................................................... 32

2.1.1 End to end Delay ............................................................................................................................... 34

2.1.2 Packet Loss ......................................................................................................................................... 36

2.1.3 Jitter ...................................................................................................................................................... 39

2.2 Playout Buffer ........................................................................................................................................... 40

2.3 Quality Model ........................................................................................................................................... 46

2.3.1 Mean Opinion Score (MOS) ............................................................................................................. 47

2.3.2 ITU-T E Model ...................................................................................................................................... 48

2.3.2.1 R factor working formula ................................................................................................................ 50

2.3.3 Correlation R factor-MOS ................................................................................................................ 53
2.4 E Model-based quality maximization algorithm .............................................. 54

Chapter 3 – Control of the Playout Delay over a Satellite Network ....................... 57
  3.1 The CNIT GEO Satellite network .................................................................. 57
  3.2 Transmission delays over Satellite Networks ............................................... 61
  3.3 Playout Buffering ....................................................................................... 64
  3.4 Measurements and Results ........................................................................ 67
    3.4.1 An overview of Adaptive Linear Filter Buffer ..................................... 69
    3.4.2 An overview of Enhanced-Normalized Least Mean Square (E-NLMS) .... 71

Chapter 4 – Control of the Playout Delay over a MANET ...................................... 74
  4.1 An Overview of VoIP in Mobile Ad hoc Networks ...................................... 75
  4.2 Proposed Strategy ..................................................................................... 79
    4.2.1 Optimized Link State Protocol (OLSR). ............................................... 81
    4.2.2 Dijkstra Algorithm ............................................................................. 84
  4.3 Playout Buffering ..................................................................................... 88
  4.4 Measurements and Results ....................................................................... 94

Chapter 5 – Conclusion and Future Works .............................................................. 98
  5.1 Unified Communication ............................................................................. 99

Acknowledgements ................................................................................................. 103

References ............................................................................................................... 104
Introduction

Intense competition is expected in the information networking arena over the next 5-10 years. As the competition increases, it will be essential for companies to position themselves appropriately to take advantage of their core competencies and to prepare for the emerging telecommunications environment. In this competitive environment, mergers, alliances, and the onslaught of new entrants into the market have service providers struggling to find innovative ways to retain and/or attract the most lucrative subscribers. Today’s service providers are striving to differentiate themselves within this expanding competitive landscape by searching for ways to brand and bundle new services, achieve operational cost reductions, and strategically position themselves in relation to their competition. Thus, many service providers are looking to Next Generation Network (NGN) services as a means to attract and/or retain the most lucrative customers. According to International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) the definition is:

“a Next generation network (NGN) is a packet-based network able to provide services including Telecommunication Services and able to make use of multiple broadband, Quality of Service-enabled transport technologies and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalized mobility which will allow consistent and ubiquitous provision of services to users.”

By this definition, an NGN can be thought of as a packet-based network where the packet switching and transport elements (e.g., routers, switches, and gateways) are logically and physically separated from the service/call control intelligence. This control intelligence is used to support all types of services over the packet-based transport network, including everything from basic voice telephony services to data, video, multimedia, advanced broadband, and management applications, which can be thought of as just another type of service that NGNs support. Although we have a feel for the types of service characteristics that will be important in an NGN environment, no one really knows what the “killer applications“ will be. Fortunately, the Next Generation Service Architecture will enable a number of key features that can be particularly beneficial to a wide array of potential services. A variety of services, some already available, others still at the conceptual stage, have been linked to NGN initiatives and considered likely candidates for NGN implementations. While some of these services can be offered on existing platforms, others benefit from the advanced control, management, and signaling capabilities of NGNs. Although emerging and new services are likely to be the strongest drivers for NGNs, most of the initial NGNs profits may actually result from the bundling of traditional services. Thus, bundled traditional services will pay for the network,
whereas emerging services and the improvement of traditional services will fuel the growth. Figure 1.1 give a brief description of several services that they will be important drivers in the NGN environment.

This thesis is focus on Voice Telephony Service in NGNs.

Since the telephone was invented in the late 1800s, telephone communication has not changed substantially. Of course, new technologies like digital circuits, DTMF (or, "touch tone"), and caller ID have improved on this invention, but the basic functionality is still the same. Over the years, service providers made a number of changes "behind the scenes" to improve on the kinds and types of services offered to subscribers, including toll-free numbers, call-return, call forwarding, etc. By and large, users do not know how those services work, but they did know two things: the same old telephone is used and the service provider charges for each and every little incremental service addition introduced.

In the 1990s, a number of individuals in research environments, both in educational and corporate institutions, took a serious interest in carrying voice and video over IP networks, especially corporate intranets and the Internet. This technology is commonly referred to today as VoIP (Voice over Internet Protocol) or IP Telephony and is, in simple terms, the process of breaking up audio or video into small packets, transmitting those chunks over an IP network, and reassembling those packets at the end point so that two people can communicate using audio and video.

One of the most important things to point out is that VoIP is not limited to voice communication as traditional communications in the Public Switched Telephone Network (PSTN). In fact, a number of efforts have been made to change this popular marketing term to better reflect the fact that VoIP means voice, video, and data conferencing. VoIP is important because, there is an opportunity to bring about significant change in the way that people
communicate. In addition to being able to use the telephones we have today to communicate in real-time, we also have the possibility of using pure IP-based phones, including desktop and wireless phones. Not only do new carriers see VoIP as a means to compete with traditional carriers, but both existing and new carriers would like to have just a single multifunctional network to carry both voice and data. The support of both voice and data traffic on a single network means lower cost. VoIP is the ability to use a single high-speed Internet connection for all voice, video, and data communications. This idea is commonly referred to as convergence and is one of the primary drivers for corporate interest in the technology. The benefit of convergence should be fairly obvious: by using a single data network for all communications, it is possible to reduce the overall maintenance and deployment costs. The benefit for both home and corporate customers is that they now have the opportunity to choose from a much larger selection of service providers to provide voice and video communication services. Since the VoIP service provider can be located virtually anywhere in the world, a person with Internet access is no longer geographically restricted in their selection of service providers and is certainly not bound to their Internet access provider.

In short, VoIP enables people to communicate in more ways and with more choices over a network using open standard-based internet protocols (the Internet).

This is both the strength and weakness of IP Telephony as the involved basic transport protocols (RTP, UDP, and IP) are not able to natively guarantee the required application Quality of Service (QoS) [3]. From the point of view of an IP Telephony Service Provider this definitely means possible waste of clients and money. Specifically the problem is at two different levels:

i) in some countries, where long distance and particularly international call tariffs are high, perhaps due to a lack of competition or due to cross subsidies to other services, the major opportunity for IP Telephony Service Providers is for price arbitrage. This means working on diffusion of an acceptable service, although not at high quality levels;

ii) in other countries, where different IP Telephony Service Providers already exist, the problem is competition for offering the best possible quality.

IP is by nature a best-effort service and offers no guarantees. For this reason, the Transmission Control Protocol (TCP) was developed as a layer above IP to ensure error-free, in-sequence delivery of information. Unfortunately, TCP provides this service at the expense of some delay, which is not a big problem for typical IP applications such as e-mail, web browsing, and file transfer. The picture changes significantly, however, when real-time applications such as voice are added to the mix. These applications are time sensitive to the extent that delays and packet loss can very easily make a service unusable. When it comes to voice, delay and jitter are of particular significance. Consequently, TCP cannot be used, which is why the User Datagram Protocol (UDP) is used instead.

Using UDP for transporting voice is fine, provided that packet loss is relatively low and the network has relatively little congestion. Traffic in an IP network can be bursty and
unpredictable, however, which can lead to a situation where one application consumes network resources for a period of time (however brief), forcing packets from other applications to wait and/or be discarded. Again, TCP can take care of this for non-real-time applications, but for other applications that use UDP, the application itself needs to take care of retransmission. Unfortunately, however, retransmission is not an option for voice because it introduces delay. If just a few packets are lost, some voice-coding schemes can handle the interruption by replaying previous packets or by extrapolating from previous packets. In the case of significant packet loss, however, voice-coding schemes will be unable to cope, which means that a speaker will have to repeat what he or she just said. Such a situation is hardly desirable, and certainly not one that a paying customer will tolerate. Therefore, it’s needed to ensure that significant delay or packet loss is avoided. If a provider wants to exploit a voice service, the quality of that voice service must at least match that provided by traditional circuit-switched network (PSTN) operators. When it comes to quality, however, circuit switching has a distinct advantage. Because a dedicated two-way transmission pipe is established between the parties in a call and because no buffering of packets or packet loss takes place, the quality is excellent. High quality is to be expected, because circuit switching was designed specifically for voice from the outset. The problem with circuit switching is that it is ill suited to other forms of communication. IP, on the other hand, is well suited to non-real-time communication and can also be made suitable for real-time communication, provided that QoS solutions are in place. A number of solutions have been developed to address the QoS issue, as IP Differentiated services (DiffServ) [4], IP Integrated services (IntServ) [5], Resource reSerVation Protocol (RSVP) [6], Multiprotocol Label Switching (MPLS) [7] and so on. These solutions approach the problem from various angles.

In this thesis and in my PhD research, I focus on the problem of jitter, which deeply affects the end-user perceived quality, jointly to delay and packet loss. Jitter is caused by the temporal variability of the network conditions. From an analysis of the jitter impairment in real-time voice communications, I proposed a new quality-based approach and I tested this on different NGNs.

The removal of transmission jitter in streaming applications is accomplished at the receive side by means of a playout buffer that masks this impairment at the expense of an additional delay. This new approach has been proposed to setting the playout buffer, which consists in taking into account the effects of delay and losses on the subjective quality. On the basis of this consideration, the playout buffering algorithm estimates the optimal buffer configuration by weighting the contribution of delay and loss to the conversational quality. The use of such a perceptually motivated optimality criterion allows the receiver to automatically balance packet delay versus packet loss. The buffer size is adaptively adjusted so that the expected quality during the next conversation period is maximized.

To estimate the quality of the voice there are different models to consider. It is increasingly important to model the VoIP speech quality. Network factors (e.g. packet loss) and source impairments (e.g. codec type) should be considered in any proposed solution. Call quality testing has traditionally been subjective: picking up a telephone and listening to the quality of
the voice. The leading subjective measurement of voice quality is the Mean Opinion Score (MOS) as described in the ITU (International Telecommunications Union) recommendation P.800 [8]. However, asking people to listen to calls over and over can be difficult and expensive to set up and execute. Considerable progress has been made in establishing objective measurements of call quality and various standards have been developed.

In my PhD studies, we have made use of the ITU-T E-Model for quality evaluation. It is a computational framework for the estimation of the conversational quality by means of a synthetic index (the R factor), which encloses the contributions of many features as delay and packet loss, presented as impairment factors.

Aim of this thesis is to improve the Quality of Service (QoS) in real-time streaming system mainly on VoIP on NGNs, and the definition of test setting for the measurement of quality parameters, considering that one of the main limitations is related to the highly variability of the network topology and channel behavior, which heavily influences the service quality due to route losses and significant delay variations. My research studies are focus on the problem of playout buffer sizing for IP Telephony services. The setting of the size of the buffer is a key operation for it directly affects the packet loss rate and the conversational interactivity, which quite influence the quality perceived by the end-user. Buffer adjustments may be introduced to avoid packet losses at all, to keep the packet loss under a desired threshold, or to have the optimal balance between loss and delay on the basis of a reference objective quality model.

The rest of this PhD thesis is organized as follows. Chapter 1 describes what is Voice over IP and explains some of its standards, Chapter 2 presents the problem of QoS in VoIP service over NGN networks, analyzing quality definitions and parameters and introducing the role of buffer in real-time communications. Chapter 3 presents a new quality-based playout algorithm that provides better results with respect to alternative approached when satellite transmissions are involved. A proposed strategy for joint routing and playout buffering over Mobile Ad Hoc Networks is described in Chapter 4. Finally, conclusions and future works are reported in Chapter 5.
Related Papers

The main contribution on which some of the chapters of this thesis are based, have already been published during the PhD studies.


- F. Boi, L. Atzori, “Quality-based technique for routing and playout buffering in IP Telephony over Mobile ad Hoc Network”, MobiMedia’08, Oulu (Finland), 7-9 July 2008.


Chapter 1 – Voice over Internet Protocol

VoIP is simply the transport of voice traffic using the Internet Protocol (IP), hardly a surprising definition. However, it is important to note that VoIP does not automatically imply voice over the Internet. The Internet is a collection of interconnected networks, all using IP. The connections between these networks are used by anyone and everyone for a wide range of applications, from e-mail to file transfer to electronic commerce (e-commerce). As we shall see later, one of the greatest challenges to VoIP is voice quality and one of the keys to acceptable voice quality is bandwidth. If we are to ensure that sufficient bandwidth is available to enable high-quality voice, then we need to control and prioritize access to the available bandwidth. Currently, that does not happen on the Internet. In fact, each user of the Internet is at the mercy of all other users. One person transferring a huge file may cause other users’ transactions to proceed more slowly. As a result, voice quality over the Internet today may vary from acceptable to atrocious. All is not lost, however. The Internet is changing rapidly in terms of its size, the number of users, and the technology that it uses. As technology changes and as more and more bandwidth is made available, it is possible that high-quality voice over the Internet may become the norm rather than the exception. However, that day is still not at hand.

Telecommunication Companies (Telcos) need standards that can improve the VoIP technology. In this chapter will be presented the most widespread standards for the management of VoIP services, and in general for real time services in NGNs. The main intention in this chapter is to provide a high-level overview of some of the aspects of particular significance to the transport of voice using IP. Then, we look at some of the issues surrounding speech coding in general and consider some of the coding standards that exist today. Finally, we describe the two main signaling protocols: H.323 and SIP.

1.1 Transport Layer: the use of UDP

When voice is to be carried on IP, it is UDP (User Datagram Protocol) [9] that is used rather than Transmission Control Protocol (TCP). TCP is a protocol than ensures ordered delivery without loss of data, while UDP is an inherently unreliable protocol. On the other hand, voice is very delay sensitive. Unfortunately, the connection setup routine in TCP and TCP’s acknowledgment routines introduce delays, which should be avoided. Even worse, in the case of lost packets, TCP will cause retransmission and thereby introduce even more delay. Tolerating
some packet loss is far better than introducing a significant delay in voice transmission. Own retransmission mechanisms are not suited to real time communications, for this reason UDP is used as transport layer (Figure 1.2). In a conversation, the occasional loss of 1 or 2 packets of voice, while not very desirable, is certainly not a catastrophe, since voice traffic will generally use small packets (10 to 40 milliseconds in duration). Obviously, modern voice-coding techniques operate better if no packets are lost, but the coding and decoding algorithms do have the capability to recover from losses, such that the occasional lost packet does not cause major degradation. So how much packet loss is tolerable? Packet loss of about 5 percent is generally acceptable, provided that the lost packets are fairly evenly spaced. A problem can arise, however, if groups of successive packets are lost.

What if packets arrive at a destination in the wrong order? If the application uses UDP, packets will be passed to the application above regardless of the order of arrival. In fact, UDP has no concept of packet order. Although these events do happen, the chances of such an event occurring during a particular session are relatively small. Furthermore, QoS management techniques can involve establishing a set path through the network for a given session, thus further reducing the probability of an out-of-sequence arrival of packets. UDP was not designed with voice traffic in mind, and voice traffic over UDP is not exactly a perfect marriage. Voice traffic over UDP just happens to be a better match than voice traffic over TCP. Nevertheless, it would be nice if some of the shortcomings of UDP could be overcome without having to resort to TCP. Fortunately, protocols exist that can help mitigate against some of the inherent weaknesses in UDP when used for real-time applications, such as voice or video.

1.2 Real time Transport Protocol (RTP)

The Real-time Transport Protocol (RTP) [10, 11] defines a standardized packet format for delivering audio and video over the Internet. It was developed by the Audio-Video Transport Working Group of the Internet Engineering Task Force (IETF) [12].

As previously discussed, UDP does nothing in terms of avoiding packet loss or even ensuring ordered delivery. RTP, which operates on top of UDP, helps address some of these functions. For example, RTP packets include a sequence number, so that the application using
RTP can at least detect the occurrence of lost packets and can ensure that received packets are presented to the user in the correct order. RTP packets also include a timestamp that corresponds to the time at which the packet was sampled from its source media stream. The destination application can use this time stamp to ensure synchronized play-out to the destination user and to calculate delay and jitter. Note that RTP itself does not do anything about these issues; rather, it simply provides additional information to a higher layer application so that the application can make reasonable decisions about how the packet of data or voice should be best handled. RTP carries the actual digitally encoded voice by taking one or more digitally encoded voice samples and attaching an RTP header so that we have RTP packets comprising an RTP header and a payload of the voice samples. These RTP packets are sent to UDP, where a UDP header is attached. The combination is then sent to IP, where an IP header is attached and the resultant IP datagram is routed to the destination. At the destination, the various headers are used to pass the packet up the stack to the appropriate application. Given that many different voice and video-coding standards exist, RTP must include a mechanism for the receiving end to know which coding standard is being used, so that the payload data can be correctly interpreted. RTP does this by including a payload type identifier in the RTP header [13].

The RTP header includes information necessary for the destination application to reconstruct the original voice sample. The RTP header is shown in Figure 1.3.

![RTP header diagram](image)

These are the most important field:

- **Marker (M):** the marker is a one-bit field, the interpretation of which is dependent upon the payload being carried. [13] states that, for an application that does not send packets during periods of silence, this bit should be set in the first packet after a period of silence (the first packet of a talkspurt). Applications that do not support silence suppression should not set this bit.

- **Payload Type (PT):** this field comprises seven bits and indicates the format of the RTP payload. In general, a single RTP packet will contain media coded according to only one payload format. Specifically, the first several octets of the payload contain information about the primary and redundant samples, including the coding schemes used, sample lengths, and timestamp data.
- **Sequence Number**: this is a 16-bit field. It is set to a random number by the sender at the beginning of a session and is incremented by one for each successive RTP packet sent. It enables the receiver to detect packet loss and/or packets arriving in the wrong order.

- **Timestamp**: this is a 32-bit field indicating the instant at which the first sample in the payload was generated. The sampling instant must be derived from a clock that increases monotonically and linearly in time so that far-end applications may play out the packets in a synchronized manner and so that jitter calculations may be performed. The resolution of the clock needs to be adequate to support synchronized playout. The clock frequency is dependent on the format of the payload data. For static payload formats, the applicable clock frequency is defined in the RTP profile. For example, the frequency for typical voice-coding schemes is 8,000 Hz. The increase in the timestamp value from packet to packet will depend on the number of samples contained in a packet. The timestamp is tied to the number of sampling instants that occur from one packet to the next. The initial value of the timestamp is a random number, chosen by the sending application.

- **Synchronization Source (SSRC)**: this is a 32-bit field indicating the synchronization source, which is the entity responsible for setting the sequence number and timestamp values and is normally the sender of the RTP packet. The identifier is chosen randomly by the sender so as not to be dependent upon network addresses. It is meant to be globally unique within an RTP session. In the case when the RTP stream is coming from a mixer, then the SSRC identifies the mixer, not the original source of media.

- **Contributing Source (CSRC)**: this is a 32-bit field containing an SSRC value for a contributor to a session. The field is used when the RTP stream comes from a mixer and is used to identify the original sources of media behind the mixer. There may be 0 to 15 CSRC entries in a single RTP header.

The RTP header is designed to accommodate the common requirements of most, if not all, media streams. Certain payload formats may, however, require additional information. This information may be contained within the payload itself, such as by specifying in the payload profile that the first n octets of the payload have some specific meaning. Alternatively, an application can apply an RTP header extension. The existence of a header extension will be indicated by setting the X bit in the RTP header to 1.

Sometimes, RTP comes with a companion control protocol, RTP Control Protocol (RTCP) [11, 12]. This protocol enables the periodic exchange of control information between session participants, with the main goal of providing quality related feedback. This feedback can be used to detect and potentially correct distribution problems. The type of information includes such details as the numbers of lost RTP packets, delays, and interarrival jitter. By using RTCP and IP multicast, a third party (such as a network operator who is not a session participant per se) can monitor session quality and detect network problems. The use of RTCP with RTP is not
mandatory. RTCP is not discussed in details because in my PhD activities it is preferred to use other techniques to quality control.

1.3 Codec

Speech coding is a very important topic. Though some analog systems still remain, they are rare. These days, voice is digitally encoded and shipped around the network (any network) as a stream of 1’s and 0’s. Speech/voice coding is simply the process by which a digital stream of 1’s and 0’s is made to represent an analog voice waveform. The process involves converting the incoming analog voice pattern into a digital stream and converting that digital stream back to an analog voice pattern at the ultimate destination. One of the reasons for implementing VoIP is the opportunity to take advantage of efficient voice coding, where fewer bits are used to represent the voice being transmitted, thereby reducing bandwidth requirements and reducing cost. The initial conclusion could be that one should implement the most bandwidth-efficient coding scheme possible, thereby saving the most money. Unfortunately, this report is not quite true. As a rough guide, the lower is the bandwidth, the lower is the quality. So if we reduce the bandwidth significantly, then we run the risk of providing substandard voice quality—not a good goal. One should not assume, however, that anything like a linear relationship exists between bandwidth and quality. Another component also affects the relationship. That is the speech-coding scheme used, which relates to processing power. For example, voice transmission at 64 Kbps sounds better than any voice transmission at 16 Kbps. Several coding schemes might exist; however, that result in a bit rate of 16 Kbps, and one of those techniques might provide better quality than another. You can be certain that the technique that provides the best quality uses a more complex algorithm and hence needs more processing power. Consequently, that technique might be more expensive to implement, emphasizing again that we get nothing for nothing. In the end, the choice of coding scheme is a balance between quality and cost. Will now be presented some of the issues surrounding speech coding in general and consider some of the coding standards that exist today. Note that this is not by any means an exhaustive analysis of coding standards. It is meant only as an overview of some of the most common standards.

Speech is generated when air is pushed from the lungs past the vocal cords and along the vocal tract. The basic vibrations occur at the vocal cords, but the sound is altered by the disposition of the vocal tract, that is, by the position of the tongue or the shape of the mouth. The vocal tract can be considered a filter and many codec technologies attempt to model the vocal tract as a filter. As the shape of the vocal tract changes relatively slowly, the transfer function of the filter needs to be changed relatively infrequently (every 20 milliseconds or so). If
the vocal tract can be modeled as a filter, then the excitation of the vibrations at the vocal cords corresponds to the excitation signal applied to the filter.

In order to create a digital representation of an analog waveform (such as voice), it is first necessary to take a number of discreet samples of the waveform and then represent each sample by some number of bits (voice sampling). The next step is the sample quantization. Now the speech signal is completely digital. These two steps (sampling and quantization) represent the coder/decoder technique, also called simply codec. Codecs basically sample and code the incoming analog signal. They then transmit quantized values of the samples to the end point, where the original signal is reconstructed.

There are many types of codecs that are different for the bandwidth and for the intrinsic quality. But we must not forget a third factor; the complexity of the compression also means a higher cost. Complex coding techniques that require little bandwidth to obtain high quality, have high complexity and therefore higher costs. We will now present a brief description of some important codecs and their encoding technique.

1.3.1 PCM Codec: G. 711

G.711 is an ITU-T standard for audio codec [14]. G.711 is the most commonplace coding technique used today. It is a waveform codec and is the coding technique that is used in circuit-switched telephone networks all over the world. G.711 has a sampling rate of 8,000 samples/second. Non-uniform quantization is used, however, with 8 bits used to represent each sample. This quantization leads to the well-known 64 Kbps digital signal 0 (DS0) rate. G.711 is often called pulse code modulation (PCM). G.711 has two variants: A-law and µ-law. µ-law is used primarily in North America and A-law is used in most other countries. The difference between the two is the manner in which non-uniform quantization is performed. Both are symmetrical around zero. A-law is skewed to be a little friendlier to lower signal levels than µ-law in that it provides very small quantization intervals for a longer range of low-level signals at the expense of larger quantization levels for a longer range of higher-level signals. Both A-law and µ-law provide very good quality. The main drawback with G.711, however, is the 64 Kbps bandwidth requirement.

1.3.2 ADPCM Codec: G.726

PCM codecs such as G.711 send individual samples to the end point, where they are reconstructed to form something close to the original waveform. Since voice changes relatively slowly, however, it is possible to predict the value of a given sample based on the values of
previous samples. In such a case, we need only transmit the difference between the predicted value and the actual value of the sample. Because the end point is performing the same predictions, it can determine the original sample value if told the difference between the prediction and the actual sample value. This technique is known as differential PCM (DPCM) and can significantly reduce the transmission bandwidth requirement without a drastic degradation in quality. In its simplest form, DPCM does not even make a prediction of the next sample; it simply transmits the difference between sample N and sample N+1, providing all the information necessary for the end point to recreate sample N+1. A slightly more advanced version of DPCM is adaptive DPCM (ADPCM). ADPCM typically predicts samples’ values based on past samples while factoring in some knowledge of how speech varies over time. The error between the actual sample and the prediction is quantized, and this quantized error is transmitted to the end point. Assuming that predictions are fairly accurate, then fewer bits are required to describe the quantized error and hence the bandwidth requirement is lower.

G.726 [15] is a more advanced ADPCM codec. A G.726 codec takes A-law- or µ-law- coded speech and converts it to or from a 16, 24, 32, or 40 Kbps channel.

Both PCM and ADPCM are waveform codecs. As such, they have effectively no algorithmic delay. In other words, the functioning of the algorithm itself does not require that quantized speech samples or quantized error values be held up for a brief time before being sent to the end point. Given that voice is a delay-sensitive form of communication, this absence of algorithm delay is a significant advantage. The disadvantage, of course, is the bandwidth that these codecs consume (especially at the higher end).

### 1.3.3 CELP Codec: G.729

Before introducing the CELP codec is necessary to step back. Previously CELP codecs, another similar codec were very successful: Analysis-by-Synthesis (AbS) codecs. Such codecs apply a simple two-state (as in Linear Predictive Coding (LPC)), voiced/unvoiced model to find the necessary input to this filter, however, the excitation signal is chosen by attempting to match the reconstructed speech waveform as closely as possible to the original speech waveform. In other words, different excitation signals are attempted, and the one that gives a result closest to the original waveform is selected (hence the name AbS). AbS codecs were first introduced in 1982 with what was to become known as the Multi-Pulse Excited (MPE) codec. Later, the Regular-Pulse Excited (RPE) and the Code-Excited Linear Predictive (CELP) codecs were introduced.

The GSM system uses Linear Predictive Coding with Regular Pulse Excitation (LPC-RPE codec). It is a full rate speech codec (GSM 06.10 Full rate [16]) and operates at 13 kbps. However, there are different versions of GSM codec, that they have different bitrate: GSM 06.20
Half rate VSELP codec (12.2 kbps) [17] and GSM 06.60 Enhanced Full rate ACELP codec (12.2 kbps)[18].

Code-Excited Linear Predictive (CELP) codecs implement a filter, the characteristics of which change over time, and they contain a codebook of acoustic vectors. Each vector contains a set of elements, where the elements represent various characteristics of the excitation signal. This approach is obviously far more sophisticated than a simple voiced/unvoiced flag. With CELP codecs, all that is transmitted to the end point is a set of information indicating filter coefficients, gain, and a pointer to the excitation vector chosen. Because the end point contains the same codebook and filter capabilities, it can reconstruct the original signal to a good degree of accuracy. There are several codecs that are differentiated by the prediction technique, but I will make a brief description of the G.729 [19], currently the most used.

G.729 is a Conjugate-Structure CELP (CS-CELP) speech codec that operates at 8 Kbps. This codec uses input frames of 10 milliseconds, corresponding to 80 samples at a sampling rate of 8,000 Hz. G.729 also includes a 5-millisecond look-ahead, resulting in an algorithmic delay of 15 milliseconds (significantly better than G.723.1). From each input frame, the codec determines linear prediction coefficients, excitation codebook indices, and gain parameters. These pieces of information are transmitted to the end point in 80-bit frames. Given that the input signal corresponds to 10 milliseconds of speech and results in a transmission of 80 bits, the transmitted bit rate is 8 Kbps. This is the standard G.729, but there are extensions, which provide rates of 6.4 kbps (Annex D, F, H, I, C+) and 11.8 kbps (Annex E, G, H, I, C+) for marginally worse and better speech quality, respectively. G.729 has been extended with various features, commonly designated as G.729a and G.729b. It’s most important the G.729 extension introduce in Annex B (G.729b) which provides a silence compression method that enables a voice activity detection (VAD) module. It is used to detect voice activity in the signal.

1.3.4 Voice Activity Detection (VAD)

Voice activity detection (VAD) is used to detect the presence of speech in an audio signal. VAD plays an important role as a preprocessing stage in numerous audio processing applications. For example, in voice over IP (VoIP) and mobile telephony applications, VAD can reduce bandwidth usage and network traffic by transmitting audio packets only if speech is detected. Furthermore, the performance of speech recognition, speaker recognition, and source localization can be improved by applying these algorithms only to parts of the audio that are identified as speech. Video conferencing is another prominent source localization application where VAD is beneficial. In such an application, source localization is performed, and the video camera is steered in the direction of the audio source when speech is detected using VAD. The typical design of a VAD algorithm is as follows [20]:

18
- There may first be a noise reduction stage, e.g. via *spectral subtraction*.
- Then some features or quantities are calculated from a section of the input signal (Figure 1.4).
- A classification rule is applied to classify the section as speech or non-speech - often this classification rule is whether the calculated value(s) exceed certain threshold(s).

![Figure 1.4. The structure of a VAD system](image)

There may be some feedback in this sequence, in which the VAD decision is used to improve the noise estimate in the noise reduction stage, or to adaptively vary the threshold(s). These feedback operations improve the VAD performance in non-stationary noise (i.e. when the noise varies a lot).

Independently from the choice of VAD algorithm, we must compromise between having voice detected as noise or noise detected as voice (between false positive and false negative). A VAD operating in a mobile phone must be able to detect speech in the presence of a range of very diverse types of acoustic background noise. In these difficult detection conditions it is often preferable that a VAD should fail-safe, indicating speech detected when the decision is in doubt, to lower the chance of losing speech segments. The biggest difficulty in the detection of speech in this environment is the very low signal-to-noise ratios (SNRs) that are encountered. It may be impossible to distinguish between speech and noise using simple level detection techniques when parts of the speech utterance are buried below the noise.

This technique is used together with the codec choice, as mentioned in the case of G.729 codec.

### 1.4 Signaling Protocol

In traditional telephony networks, specific signaling protocols are invoked before and during a call in order to communicate a desire to set up a call, to monitor call progress, and to gracefully bring a call to an end. Perhaps the best example is the ISDN User Part (ISUP), a component of the Signaling System 7 (SS7) signaling suite. In Voice over IP (VoIP) systems,
signaling protocols are necessary for exactly the same reasons. The very first VoIP systems used proprietary signaling protocols. The immediate drawback was that two users could communicate only if they both used systems from the same vendor. This lack of interoperability between systems from different vendors was a major inconvenience and impeded the early adoption of VoIP.

1.4.1 H.323: a brief description

In response to this problem, the International Telecommunications Union Telecommunications Standardization Sector (ITU-T) recommendation H.323 [21] was developed, which serves as a standardized signaling protocol for VoIP.

H.323 was designed with a good understanding of the requirements for multimedia communication over IP networks, including audio, video, and data conferencing. It defines an entire, unified system for performing these functions, leveraging the strengths of the IETF and ITU-T protocols. H.323 is a mix of several protocols and standards for performing the various phases of a connection. It borrowed much of the rich features defined in the previous generation H.320 systems that were designed for ISDN, but added to that functionality that was only possible on an IP network.

The H.323 standard specifies four kinds of components, which, when networked together, provide the point-to-point and point-to-multipoint multimedia-communication services (Figure 1.5):

- Terminals: used for real-time bidirectional multimedia communications, an H.323 terminal can either be a personal computer (PC) or a stand-alone device, running an H.323 and the multimedia applications. It supports audio communications and can optionally support video or data communications. Because the basic service provided by an H.323 terminal is audio communications, an H.323 terminal plays a key role in IP Telephony services. The primary goal of H.323 is to interwork with other multimedia terminals. H.323 terminals are compatible with H.324 terminals on Switched Circuit Network (SCN), wireless networks, and some H.3xx family terminals. H.323 terminals may be used in multipoint conferences.

- Gateways: connects two dissimilar networks. An H.323 gateway provides connectivity between an H.323 network and a non–H.323 network. For example, a gateway can connect and provide communication between an H.323 terminal and SCN networks as PSTN. This connectivity of dissimilar networks is achieved by translating protocols for call setup and release, converting media formats between different networks, and transferring information between the networks connected by the gateway. A gateway is not required, however, for communication between two terminals on an H.323 network.
- Gatekeepers: can be considered the brain of the H.323 network. It is the focal point for all calls within the H.323 network. Although they are not required, gatekeepers provide important services such as addressing, authorization and authentication of terminals and gateways; bandwidth management; accounting; billing; and charging. Gatekeepers may also provide call-routing services.

- Multipoint control units (MCUs): provide support for conferences of three or more H.323 terminals. All terminals participating in the conference establish a connection with the MCU. The MCU manages conference resources, negotiates between terminals for the purpose of determining the audio or video codec to use, and may handle the media stream. The gatekeepers, gateways, and MCUs are logically separate components of the H.323 standard but can be implemented as a single physical device.

Figure 1.5. An H.323 zone

There are several drawbacks to the H.323 protocol: its lack of scalability, binary representation for its messages, not very modular, complex signaling, large share of market, hundreds of elements, loop detection is difficult, requires full backward compatibility; summarized, it’s a very complex protocol. H.323 is mostly used in small LANs. For this reason, in my PhD studies has never been used the H.323, but we preferred to SIP protocol. The reasons will be clear after.

1.4.2 SIP: Session Initiation Protocol

Session Initiation Protocol (SIP) [22, 23] is considered a powerful alternative to H.323. They claim that SIP is a more flexible solution, simpler than H.323, easier to implement, better suited to the support of intelligent user devices, and better suited to the implementation of
advanced features. These factors are of major importance to any equipment vendor or network operator. Simplicity means that products and advanced services can be developed faster and made available to subscribers sooner. The features themselves mean that operators are better able to attract and retain customers, and they also offer the potential for new revenue streams. SIP is designed to be a part of the overall IETF multimedia data and control architecture. As such, SIP is used in conjunction with several other IETF protocols such as the Session Description Protocol (SDP), the Real-Time Streaming Protocol (RTSP), and the Session Announcement Protocol (SAP). Many developers see the SIP as the future for VoIP signaling.

SIP is a signaling protocol that handles the setup, modification, and teardown of multimedia sessions. SIP, in combination with other protocols, is used to describe the session characteristics to potential session participants. Although strictly speaking, SIP is written such that the media to be used in a given session could use any transport protocol, the media will commonly be exchanged using the Real-Time Transport Protocol (RTP) as the transport protocol. It is likely that SIP messages will pass through some of the same physical facilities as the media to be exchanged. SIP signaling should be considered separately from the media itself, however. Figure 1.6 shows the logical separation between signaling and session data. This separation is important, because the signaling may pass via one or more proxy or redirect servers while the media stream itself takes a more direct path. This approach can be considered somewhat analogous to the separation of signaling and media in H.323.

![Figure 1.6. The separation of SIP signaling and media](image)

SIP defines two basic classes of network entities: clients and servers. Strictly speaking, a client (also known as a user-agent client) is an application program that sends SIP requests. A server is an entity that responds to those requests. Thus, SIP is a client-server protocol. VoIP calls using SIP are originated by a client and terminated at a server. A client may be found within a user’s device, which could be a PC with a headset attachment or a SIP phone, for example. Clients may also be found within the same platform as a server. For example, SIP enables the use of proxies, which act as both clients and servers.

Four different types of servers exist: proxy server, redirect server, user-agent server, and registrar. Figure 1.7 gives a possible and general view of the software and hardware set up to match the requirements of the SIP.
A proxy server acts in a similar way to a proxy server used for web access from a corporate local area network (LAN). Clients send requests to the proxy and the proxy either handles those requests itself or forwards them on to other servers, perhaps after performing some translation. To those other servers, it appears as though the message is coming from the proxy rather than some entity hidden behind it. Given that a proxy both receives requests and sends requests, it incorporates both server and client functionality.

A redirect server is a server that accepts SIP requests, maps the destination address to zero or more new addresses, and returns the new address to the originator of the request. Thereafter, the originator of the request can send requests directly to the address(es) returned by the redirect server. A redirect server does not initiate any SIP requests of its own. The redirect server simply provides the information necessary to enable the originating client to correctly route the call, after which the redirect server is no longer involved.

A user-agent server accepts SIP requests and contacts the user. A response from the user to the user-agent server results in a SIP response on behalf of the user. In reality, a SIP device (such as a SIP-enabled telephone) will function as both a user-agent client and a user-agent server. Acting as a user-agent client, it is able to initiate SIP requests. Acting as a user-agent server, it is able to receive and respond to SIP requests. In practical terms, this means that it is able to initiate and receive calls. This enables SIP (a client-server protocol) to be used for peer-to-peer communication.
A registrar is a server that accepts SIP REGISTER requests. SIP includes the concept of user registration, whereby a user indicates to the network that he or she is available at a particular address. The use of registration enables SIP to support personal mobility. For example, a user could have several SIP devices, one of which might be the user’s office PC. When the user logs onto the office network, then the PC would issue a SIP REGISTER request to the appropriate registrar. Thereafter, calls can be routed to the user’s office PC. When the user leaves the office, he or she might register at a different device, such as a home PC or SIP phone. A new registration would then be performed, enabling the user to be reached at the new device.

Typically, a registrar will be combined with a proxy or redirect server. Given that practical implementations will involve the combining of a user-agent client and a user-agent server as well as the combining of registrars with either proxy servers or redirection servers, a real network may well involve only user agents and redirection or proxy servers.

1.4.2.1 SIP Session Call

At a high level, a SIP call establishment is simple. The process starts with a SIP INVITE message, which is used from calling party to called party. The message invites the called party to participate in a session, that is, a call. A number of interim responses may be made to the INVITE prior to the called party accepting the call. For example, the caller might be informed that the call is queued and/or that the called party is being alerted (the phone is ringing). Subsequently, the called party answers the calls, which generates an OK response back to the caller. The calling client acknowledges that the called party has answered by issuing an ACK message. At this point, media are exchanged. This media will most often be regular speech, but could also be other media, such as video. Finally, one of the parties hangs up, which causes a BYE message to be sent. The party receiving the BYE message sends OK to confirm receipt of the message. At that point, the call is over, as shown in Figure 1.8.
When there are multiple entities, messages remain the same, as shown in the Figure 1.9. For simplicity of presentation does not go into detail of the call establishment when there are others or all network entities.

Given that the call establishment and release is pretty straightforward, one might wonder what is so special about SIP. After all, any call-signaling protocol must have a means for one party to call another, a means to indicate acceptance of the call, and a means to release the call. SIP performs these actions and does a little more, but unlike many traditional telephony protocols, SIP offers a great deal of flexibility. For example, SIP does not care what type of media is to be exchanged during a session or the type of transport to be used for the media. In fact, SIP itself can be carried over several different transport protocols. In other words, SIP
provides more flexibility than is found in typical telecommunications protocols. That flexibility can be exploited to enable custom services and features.

SIP messages can include a number of optional fields, which can contain user-specified information. This approach enables users to share nonstandard user-specific information, which enables users and devices to make intelligent call-handling decisions. For example, a SIP INVITE can include a subject field. A person who receives an INVITE might decide to accept or reject the call depending on who is calling and what the subject happens to be. One can imagine a situation where an INVITE contains a text string such as “I know you are there. Please answer the phone.” If the call is rejected, the response might contain a text string such as “Stop calling me!” Imagine another scenario where a call is directed to a user who is currently not available. Obviously, the SIP response will indicate that the user is unavailable (no big surprise). The response could, however, include an indication that the user expects to be available again at 4:30 pm. In such a case, the calling terminal could do two things. First, the terminal could tell the calling party that the called user expects to be available at 4:30 pm. Second, at 4:30 pm, it could ask the caller if he/she wants to make the call again and, if so, the terminal can automatically set up the call. The foregoing are some simple examples of capabilities that SIP offers. Since SIP provides many pieces of information for inclusion in messages, and because additional, nonstandard information can also be included, the opportunity exists to offer numerous intelligent features to subscribers. Moreover, the control of those features is placed in the hands of the customer.

1.4.2.2 SIP Messaging Syntax: an overview

As mentioned, SIP is a signaling protocol. As such, it has a particular syntax. In the case of SIP, the syntax is text-based, using the International Organization for Standardization (ISO 10646) character set, and has a similar look and feel to the Hypertext Transfer Protocol (HTTP). One advantage of this approach is that programs designed for the parsing of HTTP can be adapted relatively easily for use with SIP. One obvious disadvantage, compared to a binary encoding, is that the messages themselves consume more bandwidth. SIP messages are either requests from a client to a server or responses (which are also known as status messages) from a server to a client. Each message, whether a request or a response, contains a start-line, possibly followed by headers and a message body:

```
message = start-line
    *message-header
    CRLF
    [message-body]
```
Given that SIP defines only request and status messages, the start-line is:

\[
\text{Start-line} = \text{request-line} \mid \text{status-line}
\]

The request-line specifies the type of request being issued, while the response line indicates the success or failure of a given request. In the case of failure, the status line indicates the type of failure or the reason for failure.

Message headers provide additional information regarding the request or response. Information that is obviously required includes the message originator and the intended message recipient. Message headers also offer the means for carrying additional information, however. For example, the Retry-after header indicates when a request should be attempted again.

The message body normally describes the type of session to be established, including a description of the media to be exchanged. Thus, for a given call, the message body might indicate that the caller wants to communicate using voice, coded according to G.711 A-law. Note, however, that SIP does not define the structure or content of the message body. This structure and content is described using a different protocol, for example by SDP. The message body could, however, contain information coded according to another standard. In fact, the message body could contain multiple parts, with each part coded according to a different format or structure. This capability is used in some situations to carry an ISDN User Part (ISUP) message in binary format, thereby enabling SIP to carry ISUP information. Carrying an ISUP message within a SIP message body would be used, for example, in a scenario where the SIP network is interworking with the Public Switched Telephone Network (PSTN) using Signaling System 7 (SS7) signaling. SIP does not greatly care about what the message body happens to say. It is concerned only with making sure that the message body is carried from one party to the other. It is at the two ends that the message body is examined. The message body can be considered to be within a sealed envelope. SIP carries it from one end to the other, but does not look inside the envelope.

A SIP request begins with a request-line, which comprises a method token, a Request-URI, and an indication of the SIP version. The method token identifies the specific request being issued and the Request-URI is the address of the entity to which the request is being sent. The three components of the request-line are separated by spaces, and a CRLF character terminates the line itself. Thus, the syntax is as follows:

\[
\text{request-line} = \text{method SP Request-URI SP SIP-Version CRLF}
\]

RFC 3261 [23] defines six different methods (and hence six different types of requests): INVITE, ACK, OPTIONS, BYE, CANCEL, and REGISTER. A number of extensions to SIP specify additional methods, such as INFO, REFER, and UPDATE.
The start line of a SIP response is a status line. This contains a status code, which is a three-digit number indicating the outcome of the request. The start line will also contain a reason phrase, which provides a textual description of the outcome. The reason code will be interpreted and acted upon by the client software, while the reason-phrase could be presented to the human user to aid in understanding the response. The syntax of the status line is:

```
status-line = SIP version SP status code SP reason-phrase CRLF
```

Status codes defined in SIP have values between 100 and 699, with the first digit of the reason code indicating the class of response as follows. Thus, all status codes between 100 and 199 belong to the same class.

I don’t go into details of the request/response messages because this overview will only show the simplicity of SIP.

As with any signaling protocol, requests and responses are sent to particular addresses. In SIP, these addresses are known as SIP Uniform Resource Indicators (URIs). These addresses take the form of user@host, which is very similar to an e-mail address. In many cases, a given user’s SIP address can, in fact, be guessed from the user’s e-mail address. Although the two may look similar, however, they are different. Whereas an e-mail address uses a mailto Uniform Resource Locator (URL), such as mailto:boi@unica.it, a SIP URI has the syntax sip:boi@unica.it. SIP deals with multimedia sessions, which could include voice. Furthermore, SIP networks need to interwork with traditional circuit-switched networks. For these reasons, SIP enables the user portion of the IP address to be a telephone number. Thus, we could have a SIP address such as sip:24011978@unica.it. In a given network, such a SIP address could be used to cause media to be routed to a gateway that interfaces with the traditional telephone company. A SIP URI may also be supplemented by a number of parameters that provide more information. For example, to clearly indicate that a call is to a telephone number, the URI could be supplemented by the term user=phone. In such a case, the URI would have the format sip:24011978@unica.it;user=phone.

Figure 1.10 shows an example registration scenario, where the syntax of message sequence is shown.
1.4.2.3 **SIP Interworking**

Obviously, SIP-based networks will never be the only types of networks in existence. To begin with, circuit-switched networks will continue to be with us for a very long time. Therefore, an obvious need exists for SIP-based networks to be able to interwork with the circuit-switched networks of the PSTN. Furthermore, although SIP is viewed by many as the future of IP telephony, an embedded base of H.323 systems already exists, and more H.323 systems are being deployed. Therefore, a need also exists for SIP based networks to interwork with H.323-based networks. The SIP specification, however, does not specifically address how such interworking should be achieved.

For interworking with the PSTN, gateways will be required to provide the conversion from circuit-switched media to the packet and vice versa. Not only must there be media interworking, but there must also be signaling interworking. After all, SIP is a signaling protocol. If calls are to be established between a SIP-based network and the PSTN, then the SIP network must be able to communicate with the PSTN according to the signaling protocol used in the PSTN. In most cases, this protocol is SS7. For the establishment, maintenance, and teardown of speech calls, the applicable SS7 protocol is the ISDN User Part (ISUP).

It is sufficient to assume that a gateway entity exists on the network for interworking between SIP and the PSTN. Let us call that entity a network gateway (NGW). In such a case, let us consider a call from Manager (at work) to User (at home), where User has a standard
residential phone connected to a PSTN switch. Such a scenario would appear as in Figure 1.11. Note that most of the fields within the SIP messages and responses are omitted to avoid clutter.

Recall that SIP enables a SIP URI to have the form of a telephone number. The INVITE includes such a URI in the To: field, which is mapped to the calling party number of the initial address message (IAM). The From: field of the INVITE could also have a URI in the form of a telephone for the calling party. This would enable the called party to still avail itself of the PSTN Caller-ID service. The PSTN switch responds with an ACM, which is mapped to the SIP 183 (session progress) response. This response contains a session description and enables in-band information to be returned from the called switch to the SIP caller. Once the call is answered, an ISUP ANM is returned, which is mapped to the 200 (OK) response. This response will have an updated session description, since the session description in the 183 response is considered temporary and is part of an early dialog.

If the call were to be in the opposite direction, from the PSTN to a SIP user, the 180 ringing response is used instead of the 183 session progress response, but this is for example purposes.
only. In this example, the 180 ringing response is used by the NGW to trigger a ringing tone towards the calling party locally. Alternatively, the called SIP user agent might return a 183 (session progress) response, with an early media description (such as a ringing tone) to be played to the caller. It can be assumed that the reliability of these responses would be applied, so that PRACK methods and corresponding 200 (OK) responses would occur.

Although this interworking example might seem quite straightforward, seamless interworking between two different protocols is not often quite that easy, because rarely does a one-to-one mapping of messages and parameters take place between one protocol and the other. Sometimes the mapping from protocol A to B means that what is sent in protocol B is only an approximation of what was received in protocol A. If dual interworking is used (from protocol A to protocol B and back to protocol A), what comes out at one end may not be the same as what went in at the other end.

Given that certain VoIP networks provide a transit service from PSTN at one end to PSTN at the other end, this lack of transparency is a concern. The solution to this issue is known as SIP for Telephony (SIP-T), which is documented in an Internet draft. SIP-T is not so much an extension to SIP, but rather it is a specification of how best to use SIP when interworking with traditional telephony networks. As such, the SIP-T draft is categorized as a Best Current Practice (BCP) rather than a potential technical standard.

A significant aspect of SIP-T is the capability to carry ISUP or QSIG message contents within a SIP message body. This capability is described in RFC 3204 [24]. The approach is to carry an ISUP or QSIG message within a SIP message body in hex format. In such a case, the SIP message body could be a multipart message body, including both an SDP session description as one part and an ISUP or QSIG message as another. The SIP INFO method is another component of SIP-T, as it provides for the transfer of midcall information within a SIP session. Such midcall information would typically be generated within the PSTN.
Chapter 2 – QoS in VoIP services

To many, quality of service (QoS) is one of the biggest, if not the biggest, issue facing Voice over IP (VoIP). QoS (or the lack thereof) is one reason why many of the VoIP solutions on the market today that provide voice over the Internet are free services. The philosophy behind such offerings is that customers can hardly complain about poor quality when they are being provided with a free service. This is one reason that drives us to do research to improve the QoS, in addition to being a strong business for Telcos.

Not only do new carriers see VoIP as a means to compete with traditional carriers, but both existing and new carriers would like to have just a single multifunctional network to carry both voice and data. The support of both voice and data traffic on a single network means lower cost. Again, however, the challenge is in ensuring a high-quality service and making sure that one type of traffic does not consume excessive network resources to the detriment of other types of traffic.

This chapter focuses on a range of technical solutions to improve QoS in a VoIP network. First, however, it is worth considering what QoS means and why it is a question for VoIP networks. After presenting in detail QoS problems faced in VoIP that worsen the quality of application, quality evaluation models will be evaluated: MOS and E Model. Then, it will analyze the issue of playout delay, with the presentation of some techniques for playout buffering. In particular, we present a quality-based playout buffer that is the subject of my PhD studies.

2.1 Key Performance Indicators for QoS

QoS is a collective measure of the level of service delivered to a customer. It can be considered as the level of assurance from a particular application that the network can meet its service requirements. From a technical perspective, QoS can be characterized by several performance criteria such as availability (low downtime), throughput, connection setup time, the percentage of successful transmissions, and the speed of fault detection and correction. In an Internet Protocol (IP) network, QoS can be measured in terms of bandwidth, packet loss, delay, and jitter. In order to provide a high QoS, the IP network needs to provide assurances that, for a given session or a set of sessions, the measurement of these characteristics will fall within certain bounds.
QoS is not just a technical matter on just one operator’s network. In general, a customer does not know or care about how a provider gets a call from one place to another, provided that the call gets there and meets the customer’s quality expectations. This means that QoS must be end to end and requires the support of all networks in the chain. A call might originate on one provider’s network and terminate on another provider’s network, perhaps traversing other networks in between. Each of the networks must cooperate to ensure that the quality provided matches what is expected. This requirement raises the issue of Service Level Agreements (SLAs) between different operators. SLAs are agreements in which operators make commitments to each other regarding the type and QoS to be offered to each other and the penalties involved if such commitments are not met. SLAs at least offer the potential of ensuring quality across multiple networks. The SLA requirements of a service need to be derived from the SLA requirements of the applications they are intended to support; customers utilizing the service rely on this contract to ensure that they can deliver the applications critical to their business. Hence, SLA definitions are key and it is essential they are representative of the characteristics of the IP transport service they define. Application and service SLA requirements are the inputs and also the qualification criteria for measuring success in a network QoS design; a network which provides a 500 ms one-way delay would clearly not be able to support a VoIP service requiring a worst-case one-way delay of about 200 ms. Similarly, a network that provides a one-way delay of 50 ms may be over-engineered to support this service, and over-engineering may incur unnecessary cost. In price-sensitive markets, whether customers will be prepared to pay for the facility that QoS provides may depend in part on whether they can detect the effects of QoS; SLAs can provide a means to qualify the difference between services. In considering SLAs and SLA metrics, because they define a service contract, as with any contract the detail of the contract definition matters. In terms of SLAs for IP service performance, it is important to understand how the SLAs are numerically defined; SLAs may be defined in absolute terms, e.g. a worst-case one-way delay of 100 ms, or may be defined statistically, e.g. a loss rate of 0.01%. In the case of the statistical definition, defining a network loss rate of 0.01% is not sufficient information on its own to be able to determine if an application or service could be supported on that network. How the loss rate is measured and calculated needs to be defined in order to understand what impact the 0.01% loss rate will have on the end applications; 1 lost packet in every one hundred packets may not have a significant impact on a VoIP call, but 10 consecutive packets dropped out of 1000 will cause a glitch in the call that is audible to the end-user.

Different applications have different SLA requirements; the impact that different network services with different SLAs have on an application is dependent upon the specific application, for example: excessive packet loss or delay may make it difficult to support real-time applications although the precise threshold of “excessive” depends on the particular application, as VoIP.

The key performance indicators (KPIs) that determine the impact that variations in networks SLA characteristics such as delay and loss have on VoIP are the codec that is used to
encode the signal and the specific details of the end-system implementation. For example, some codecs may be less tolerant to loss than others, while a poor end-system implementation may be less tolerant to jitter.

The following sub-sections will introduce the 3 main network-related KPIs of VoIP: delay, loss and jitter.

2.1.1 End to end Delay

For VoIP the important delay metric is the end to end delay, in each direction. End to end delay (also named latency) is the time delay incurred in speech by the IP Telephony system. Delay is typically measured in milliseconds (ms) from the moment that the speaker produces a speech until the listener hears that speech. This is termed as "mouth-to-ear" delay or the "one-way" delay. The main impact that end to end delay has on VoIP is to the interactivity of conversational speech. If the delay is too high, participants find it difficult to discern the difference between natural pauses in speech and the delays introduced by the system. If they mistake system delays for pauses in conversation and take these delays as their cue to begin to speak, by the time their words arrive at the other end, the other speaker may have already started to speak with the result that the normal protocol of conversation breaks down. Excessive end to end delay can also impair the effectiveness of mechanisms used for echo-cancellation. In the traditional PSTN and in VoIP, the round-trip latency for domestic calls is virtually always under 150 ms. In VoIP services, higher delays may also be acceptable, but with a consequent reduction in user satisfaction, with delays exceeding 400 ms generally considered unacceptable. A set of delay specifications from Cisco Systems is provided in Table 2.1, while subjective (or perceived) quality is directly linked to the delay as appears in Figure 2.1.

Table 2.1. A set of delay specifications by Cisco Systems
Network delay is only one component of the one-way delay that impacts a VoIP call. Hence, having determined what the maximum acceptable mouth-to-ear delay is for a particular VoIP service, a network QOS design should take this budget and apportion it to the various components of network delay (propagation delay through the backbone, scheduling delay due to congestion, and the access link serialization delay) and end-system delay (due to VoIP codec and de-jitter buffer). The example timeline in Figure 2.2 shows the components of delay, which impact the mouth-to-ear delay of a VoIP service, using typical values for each component.

The codec delays are dependent upon the type of codec used. The Table 2.2 lists the maximum theoretical one-way delay introduced by codec-related processing for some of the more common VoIP codecs; in practice VoIP end-systems may incur an additional 5–20 ms of delay, depending upon the specific implementation. One-way network delays of 35–100 ms are
typically targeted for high quality ("toll quality") VoIP services, in order to ensure that a one-way delay of 150 ms can be achieved. Higher delays may be tolerated for lower quality services.

Table 2.2. Theoretical one-way delay introduced by some of the more common VoIP codecs

<table>
<thead>
<tr>
<th>ITU-T Codec</th>
<th>Codec type</th>
<th>Maximum codec delay (ms) (a1 d)</th>
<th>Bitrate (bps)</th>
<th>Packetization interval (ms) (b)</th>
<th>pps</th>
<th>Payload size (bytes)</th>
<th>IP pkt size (bytes)</th>
<th>IP bps</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>PCM</td>
<td>0.375</td>
<td>64000</td>
<td>20</td>
<td>100</td>
<td>80</td>
<td>80</td>
<td>120</td>
</tr>
<tr>
<td>G.711</td>
<td>PCM</td>
<td>0.375</td>
<td>64000</td>
<td>30</td>
<td>50</td>
<td>160</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>G.711</td>
<td>PCM</td>
<td>0.375</td>
<td>64000</td>
<td>30</td>
<td>33.33</td>
<td>240</td>
<td>280</td>
<td>280</td>
</tr>
<tr>
<td>G.723.1</td>
<td>ACELP</td>
<td>97.5</td>
<td>5300</td>
<td>30</td>
<td>33.33</td>
<td>20</td>
<td>60</td>
<td>15000</td>
</tr>
<tr>
<td>G.723.1</td>
<td>ACELP</td>
<td>97.5</td>
<td>5300</td>
<td>15</td>
<td>16.67</td>
<td>40</td>
<td>80</td>
<td>10000</td>
</tr>
<tr>
<td>G.726.16</td>
<td>ADPCM</td>
<td>0.375</td>
<td>16000</td>
<td>10</td>
<td>100</td>
<td>20</td>
<td>60</td>
<td>40000</td>
</tr>
<tr>
<td>G.726.16</td>
<td>ADPCM</td>
<td>0.375</td>
<td>16000</td>
<td>20</td>
<td>50</td>
<td>40</td>
<td>80</td>
<td>32000</td>
</tr>
<tr>
<td>G.726.16</td>
<td>ADPCM</td>
<td>0.375</td>
<td>16000</td>
<td>30</td>
<td>33.33</td>
<td>60</td>
<td>100</td>
<td>26664</td>
</tr>
<tr>
<td>G.726.24</td>
<td>ADPCM</td>
<td>0.375</td>
<td>24000</td>
<td>10</td>
<td>100</td>
<td>30</td>
<td>70</td>
<td>56000</td>
</tr>
<tr>
<td>G.726.24</td>
<td>ADPCM</td>
<td>0.375</td>
<td>24000</td>
<td>20</td>
<td>50</td>
<td>60</td>
<td>100</td>
<td>40000</td>
</tr>
<tr>
<td>G.726.24</td>
<td>ADPCM</td>
<td>0.375</td>
<td>24000</td>
<td>30</td>
<td>33.33</td>
<td>90</td>
<td>130</td>
<td>34663</td>
</tr>
<tr>
<td>G.726.32</td>
<td>ADPCM</td>
<td>0.375</td>
<td>32000</td>
<td>10</td>
<td>100</td>
<td>40</td>
<td>80</td>
<td>64000</td>
</tr>
<tr>
<td>G.726.32</td>
<td>ADPCM</td>
<td>0.375</td>
<td>32000</td>
<td>20</td>
<td>50</td>
<td>80</td>
<td>120</td>
<td>48000</td>
</tr>
<tr>
<td>G.726.32</td>
<td>ADPCM</td>
<td>0.375</td>
<td>32000</td>
<td>30</td>
<td>33.33</td>
<td>120</td>
<td>160</td>
<td>42662</td>
</tr>
<tr>
<td>G.726.40</td>
<td>ADPCM</td>
<td>0.375</td>
<td>40000</td>
<td>10</td>
<td>100</td>
<td>50</td>
<td>90</td>
<td>72000</td>
</tr>
<tr>
<td>G.726.40</td>
<td>ADPCM</td>
<td>0.375</td>
<td>40000</td>
<td>20</td>
<td>50</td>
<td>100</td>
<td>140</td>
<td>56000</td>
</tr>
<tr>
<td>G.726.40</td>
<td>ADPCM</td>
<td>0.375</td>
<td>40000</td>
<td>30</td>
<td>33.33</td>
<td>150</td>
<td>190</td>
<td>50662</td>
</tr>
<tr>
<td>G.728</td>
<td>LD-CELP</td>
<td>1.875</td>
<td>16000</td>
<td>10</td>
<td>100</td>
<td>20</td>
<td>60</td>
<td>48000</td>
</tr>
<tr>
<td>G.728</td>
<td>LD-CELP</td>
<td>1.875</td>
<td>16000</td>
<td>20</td>
<td>50</td>
<td>40</td>
<td>80</td>
<td>32000</td>
</tr>
<tr>
<td>G.729A</td>
<td>CS-ACELP</td>
<td>35</td>
<td>8000</td>
<td>10</td>
<td>100</td>
<td>10</td>
<td>40</td>
<td>200</td>
</tr>
<tr>
<td>G.729A</td>
<td>CS-ACELP</td>
<td>35</td>
<td>8000</td>
<td>20</td>
<td>50</td>
<td>20</td>
<td>60</td>
<td>24000</td>
</tr>
</tbody>
</table>

2.1.2 Packet Loss

The loss event is always related to network. An IP network presents some short-term correlation features, that is if the n\textsuperscript{th} packet is lost also the (n+1)\textsuperscript{th} packet could be lost. This phenomenon causes the loss burstiness. Apart from the overall percentage of lost packets, the burstiness is important, in particular, for a correct estimation of the speech quality. In fact, approximating bursty losses as they would be random implies an overestimation of speech quality degradation. The loss is usually modeled by means of a 2-state Gilbert model [25]. Usually the threshold of overall packet loss rate for a good quality is fixed to 5%. This means that for greater values of overall loss the perceived quality is unacceptable. Packet loss can be caused by a number of factors:
**Congestion.** When congestion occurs, queues build up and packets are dropped. Loss due to congestion is controlled by managing the traffic load and by applying appropriate queuing and scheduling mechanisms.

**Lower layer errors.** Physical layer bit errors, which may be due to noise or attenuation in the transmission channel, may cause packets to be dropped. Most link layer technologies and IP transport protocols, such as UDP, have a cyclic redundancy check (CRC) or parity checksum to detect bit errors; when bit errors occur and the checksum is incorrect, the impacted frames will be dropped. Hence, for packets traversing networks with such capabilities, bit errors will normally result in packet loss, i.e. each packet will either arrive correct or not at all, although there are a few noted exceptions to this. In practice, actual bit error rates (BER, also referred to as the bit error ratio) vary widely depending upon the underlying layer 1 or layer 2 technologies used, which is different for different parts of the network, as Fiber-based optical, SDH or others links. For link layer technologies that are generally prone to high error rates, it is usual to support some link layer reliability mechanisms, such as Forward Error Correction (FEC), in order to recover from some bit error cases. If, however, the underlying layer 1 or layer 2 technologies cannot provide the BERs necessary to support the packet loss rates (PLRs) required by IP applications, then error correction or concealment techniques need to be used either by higher layer protocols or by the application, or alternate layer 1 or layer 2 technologies are needed.

**Network element failures.** Network element failures may cause packets to be dropped until connectively is restored around the failed network element. The resulting loss period depends upon the underlying network technologies that are used. With a “plain” IP (i.e. non-MPLS) deployment, after a network element failure, even if there is an alternative path around the failure, there will be a loss of connectivity which causes packet loss until the interior gateway routing protocol (IGP) converges. In well-designed networks, the IGP convergence time completes in a few hundred milliseconds [26]. If there is not an alternative path available then the loss of connectivity will persist until the failure is repaired. While such outages could be accounted for by the defined loss rate for the service, they are most commonly accounted for in the defined availability for the service. Where an alternate path exists, the loss of connectivity following network element failures can be significantly reduced through the use of technologies such as MPLS Traffic Engineering (TE) Fast Reroute (FRR) or IP Fast Reroute (IPFRR), which are local protection techniques that enable connectivity to be rapidly restored around link and node failures, typically within 50 ms. Equivalent techniques may be employed at layer 2, such as Automatic Protection Switching (APS) for SONET and Multiplex Section Protection (MSP) for SDH.
Loss in application end-systems. Loss in application end-systems can happen due to overflows and underflows in the receiving buffer. An overflow is where the buffer is already full and another packet arrives, which cannot therefore be enqueued in the buffer; overflows can potentially impact all types of applications. An underflow typically only impacts real-time applications, such as VoIP and video, and is where the buffer is empty when the codec needs to play out a sample, and is effectively realized as a “lost” packet. Loss due to buffer underflows and overflows can be prevented through careful design both of the network and the application end-systems. Playout buffer size is the main problem that will be addressed later.

Depending upon the transport protocol or application, there are potentially a number of techniques that can be employed to protect against packet loss including error correction, error concealment, redundant transmission/retransmission and others technique to adjust the playout buffer, that is the focus of this PhD thesis.

For example, Packet Loss Concealment (PLC) is a technique used to mask the effects of lost or discarded VoIP packets; an understanding of PLC is needed to understand the impact that packet loss has on the resultant quality of a VoIP call. The method of packet loss concealment used depends upon the type of codec used. A simple method of packet loss concealment, used by waveform codecs like G.711, is to replay the previously received sample; the concept underlying this approach is that, except for rapidly changing sections, the speech signal is locally stationary. This technique can be effective at concealing the loss of up to approximately 20 ms of samples. The packetization interval determines the size of samples contained within a single packet; assuming a 20 ms packetization interval, the loss of two or more consecutive packets will result in a noticeable degradation of voice quality, as mentioned. From a network design perspective, it is important to note that a design decision to use a 30 ms packetization interval, for a given probability of packet loss, could result in worse perceived call quality than a 20 ms packetization interval, as with a 30 ms interval PLC may not be able to effectively conceal the loss of a single packet. Hence, there is a network design trade-off to be considered; larger packetization intervals may reduce the bandwidth of a VoIP call (there is less IP overhead due to more samples being carried in a single packet) but may also result in lower call quality for a given loss rate.

Low bit rate frame-based codecs, such as G.729 and G.723, use more sophisticated PLC techniques, which can conceal up to 30–40 ms of loss with “tolerable” quality, when the available history used for the interpolation is still relevant. Concealment becomes problematic with short phonemes (the smallest phonetic unit in a language) where 30 ms of samples can be over half of a phoneme and previous and subsequent samples may not provide enough information about the lost sample to allow it to be effectively concealed. With frame-based codecs, the packetization interval will determine the number of frames carried in a single packet. Similarly as for waveform-based codecs, if the packetization interval were greater than the loss that the PLC algorithm can interpolate for, then PLC would not be able to conceal the
loss of a single packet effectively. Hence, to summarize the impact that packet loss has on VoIP, with an appropriately selected packetization interval (20–30 ms depending upon the type of codec used) a loss period of one packet may be concealed but a loss period of two or more consecutive packets may result in a noticeable degradation of voice quality. The loss distance targeted for a particular service is a choice for the service provider.

Therefore, in practice, networks supporting VoIP should typically be designed for very close to zero percent VoIP packet loss. QoS mechanisms, admission control techniques and appropriate capacity planning techniques are deployed to ensure that no packets are lost due to congestion with the only actual packet loss being due to layer 1 bit errors or network element failures. Where packet loss occurs, the impact of the loss should be reduced to acceptable levels using PLC techniques or/and using special techniques for playout buffer oriented to QoS maximization, as the technique that will be presented later.

2.1.3 Jitter

The jitter is caused by the high variability of the network component of the one-way delay. It could be related to some time-variant network factors, such as network congestion, improper queuing, or configuration errors. The result is that the destination receives a flow of packets with irregular inter-packet delays. This point is better explained in the following. At the sending side the instants of generation of the speech packets are k*T, where k is the number of the emitted transport packet (0…n) while T depends on the number and size of the speech frames conveyed in every transport packet (e.g., for the ITU-T G.729 speech codec, usually T = 20 ms, corresponding to 2 speech frames of 10 ms). When these packets are sent to the network their order can change due to network variable conditions. This causes an alteration in the order of arrival of transport packet: as an example, packet X can arrive later than X+1. Thus, a buffer at the receiver is needed to reconstruct the right order and guarantee a good decoding. The usage of this buffer is at the base of a playout control mechanism. The setting of the size of the buffer is a key operation in IP Telephony as it directly affects the packet loss rate and the conversational interactivity, thus influencing the quality perceived by the end-user, in fact, as shown in Figure 2.3, the 3rd packet don’t come and the playout buffer cannot process and lose it. Buffer adjustments may be introduced to avoid packet losses at all, to keep the packet loss under a desired threshold, or to have the optimal balance between loss and delay on the basis of a reference quality model. Not only is the absolute size value important, but when it is modified: the common attitude is to avoid buffer adjustments during the periods of speech.

Jitter is caused in IP networks in two ways. First, packets can take different routes from sender to receiver and consequently experience different delays. Second, a given packet in a voice conversation can experience longer queuing times than the previous packet-even if they
both traveled the same route. This is due to the fact that the network resources, and particularly queues within routers, might be used by other traffic as well as voice traffic. In contrast to the situation in IP networks, jitter is not an issue in circuit switched networks, because an open, dedicated pipe exists from the sender to the receiver for the duration of the conversation. Thus, all the speech follows the same path and does not have to share resources with any other traffic. What goes in one side comes out the other side after a fixed delay.

The behavior of the playout buffer will be explained in details in the next section.

![Diagram of Jitter Phenomenon](image)

**Figure 2.3. Jitter phenomenon**

### 2.2 Playout Buffer

The jitter phenomenon is sketched in Figure 2.3 in the middle row of packets (note that some packets may also be missing due to network losses, as happened to packet 3). The jitter is quite harmful since the decoder, to avoid the degradation of the listening quality, should play out the data smoothly (that is, with the same interval $T$ of the sender side). To this aim, a playout buffer is used at the receive-side to compensate the transmission jitter with the addition of additional delay in the end-to-end chain. The intent is to re-obtain a sequence of playout instants uniformly spaced, as illustrated in the bottom row of the figure (i.e., the distance between each couple of consecutive packets should be the same as during voice coding). Accordingly, the delay between the departure and the decoding, called end-to-end delay or playout delay, is kept equal for every packet (at least for a group of packets, usually corresponding to an entire talkspurt) and given by the sum of the buffering and the network delays, which instead are both random. The setting of the playout delay for the first packet is
shown in Figure 2.3. Usually, the receiving end discards the packets that arrive later than their scheduled instants (playout instants) and tries to conceal these losses usually predicting the missing parameters from the previously received packets. Less often, it is not discarded, but the previous packet content is stretched till the arrival of the late packet so as not to stop the playout.

If the dejitter buffer playout delay is set either arbitrarily large or arbitrarily small, then it may impose unnecessary constraints on the characteristics of the network or may affect the quality of the VoIP service. A playout delay set too large adds unnecessarily to the end-to-end delay, as shown in Figure 2.4, meaning that less of the ear-to-mouth delay budget is available to be apportioned to the network and hence that the network needs to support a tighter delay target than practically necessary.

![Figure 2.4. Playout delay too large](image)

If the dejitter buffer playout delay is too small to accommodate the network jitter then buffer underflows can occur; an underflow is where the buffer is empty when the codec needs to play out a sample, and is effectively realized as a “lost” packet as shown in Figure 2.5.

Most VoIP end-systems use adaptive dejitter buffers, which aim to overcome these issues by dynamically tuning the playout delay to the lowest acceptable value, as shown in Figure 2.6.

Well-designed adaptive dejitter buffer algorithms should not impose any unnecessary constraints on the network design.
The playout algorithms are devoted to setting the end-to-end delay, which is changed during the streaming session as described in the following.

The problem of jitter compensation has been addressed by using several approaches since the early 80’s. [27] and [28] are some of the first works that propose the introduction of additional delay to solve the problem of playout synchronization between sender and receiver. Since then, many other strategies have been investigated. In the following, we propose a classification that is summarized in Figure 2.7. At first, two groups are introduced, that is, those of fixed and adaptive approaches.
According to the techniques belonging to the first group, the end-to-end delay is kept constant for all voice packets in a session [29, 27]. Such an approach is inefficient, since a fixed configuration is not reliable with the temporal variability of the network behavior. As shown in Figure 2.8, where the points represent the RTP packets and rectangles are the moments of voice called talkspurt interspersed by moments of silence, many RTP packets are lost because of the value of fixed playout that does not fit the variables of the network.

![Figure 2.7. A classification of playout strategies](image)

According to the techniques belonging to the first group, the end-to-end delay is kept constant for all voice packets in a session [29, 27]. Such an approach is inefficient, since a fixed configuration is not reliable with the temporal variability of the network behavior. As shown in Figure 2.8, where the points represent the RTP packets and rectangles are the moments of voice called talkspurt interspersed by moments of silence, many RTP packets are lost because of the value of fixed playout that does not fit the variables of the network.

![Figure 2.8. Fixed playout strategy](image)

Differently, techniques in the second group work adapting the delay to the changing network conditions, preventing from using high delays during low congestion conditions and vice versa. In this case, the network behavior is monitored during the streaming session and the buffer size adjusted accordingly. One of the main features of the adaptive algorithms is when the playout buffer is adjusted. Intra-talkspurt techniques modify the end-to-end delay independently from the silence periods, using some strategies of compression and extension of the waveform, as shown in Figure 2.9.

![Figure 2.9. Intra-talkspurt techniques](image)
On the other hand, between-talkspurt methods act during the intervals of silence, as shown in Figure 2.10. The later methods are more frequently used since they don’t require any signal processing techniques to change the length of the speech. On the contrary, they are not able to take into account high-delay spikes within talkspurts, resulting occasionally in a burst of losses and audible quality degradation.

Among the adaptive between-talkspurt techniques, some estimate the network delay statistics and set the playout delay so that only a small fraction of packets arrive late. This approach, which we call loss-intolerant, does not take into account any loss concealment, resulting in an overestimation of the required the playout delay. Indeed, VoIP applications can tolerate or conceal a certain amount of late packets; accordingly, some playout techniques perform a controlled tradeoff between the packet loss and the delay. A percentage of packet loss is then allowed and the playout delay is set so as to reach this “target value”. We refer to these as loss-tolerant techniques. Another category of algorithms is named quality-based; these are driven by the maximization of a metric linked to the end-user perceived quality. These algorithms appeared in the literature only few years ago, and have been triggered by the advancements reached by the standardization committees working on the development of voice QoS evaluation tools within ITU [30, 31], ETSI [32, 33], and ANSI [34] organizations.

In this PhD thesis focus mainly on the between-talkspurts strategies, more frequently used for computational complexity reasons. In the following the loss-intolerant, loss-tolerant and
quality-based strategies are presented together. The idea is not to give a detailed explanation but only the indication of the state of art within the indicated categories that is the basis of my research. The first category is the Loss-intolerant. Such techniques work between talkspurts; it means that changes in the playout delay are introduced only during silence periods not requiring the decoder to perform waveform adjustments. The underlying idea is that packet losses are undesired in a voice packet communication and should be avoided even at the expense of high end-to-end delays (not considering an eventual FEC system). One of the first works based on this approach is by Ramjee et al. [35], who proposed the use of a playout delay high enough to expect only a small fraction of packet losses. Essentially, the playout delay is estimated making use of a linear recursive filter characterized by a fixed weighting factor $\alpha$. But, choosing a fixed $\alpha$ may be inappropriate under typical present-day Internet conditions. For this reason, several variants of this algorithm have been proposed. The $\alpha$-Adaptive technique [36] generalizes the filtering method in [35] by using several values of $\alpha$. The adaptation is performed by looking at the resulting packet loss rate during the past in case of using the current $\alpha$ value minus/plus $\Delta$; the correct value of $\alpha$ is then chosen. Differently, in [37], the application of a Normalized Least Mean Square (NLMS) active predictor is proposed. This is used to estimate the network delay for each packet from the previous N ones. [38] is the most recent work based on the loss-intolerant approach, that is based on a dynamic adjustment of $\alpha$. The main advantage of the loss-intolerant approach is to be straightforward and then simple to be implemented. I generally believe that this approach has to be employed when the user is aware of the network condition which should be stable so as not to have too high end-to-end delays. Loss-intolerant algorithms in last section estimate an average network delay and use it to set the playout delay so that the fraction of late packets is kept very small. However, IP Telephony applications can tolerate or conceal a small amount of late packets after the playout buffer. On the basis of this observation, some strategies have been devised. They are called Loss-tolerant.

The Concord algorithm in [39] is maybe one of the most known strategies. It fixes two thresholds for the maximum late packet percentage and maximum acceptable delay. The strategy makes use of a histogram of the network delay observed for past packets; from the histogram, an approximated and sampled version of the packet delay distribution (PDD) is computed. By using such PDD, the algorithm set the playout delay so that the two constraints on the maximum late packets percentage and maximum acceptable delay are satisfied. Other good examples of Loss-tolerant strategies are presented in [40, 41], and basically differ from the concord in a way the PDD is updated and considered. The authors suggest also a reading of the work presented in [42], whom main contribution consists in the evaluation of the interactions between Forward Error Correction (FEC) and playout buffering. In particular, the authors evaluate the repair capability of the FEC scheme and its influence on both the total loss percentage and the network delay. The last category is the quality-based approach. The E-MOS algorithm in [43] is the one of the first playout buffer techniques that tries the way of using quality metrics. This work relies on the studies by Savolaine in [44], who provides the
experimental relationships between MOS [8] and the playout delay at different packet loss rates for the ITU-T G.711 speech codec. The authors strongly suggest also the work presented in [44]. In [45], the joint problem of FEC and playout buffering is addressed. The quality is evaluated making use of the ITU-T E-Model with some additional elaborations. One of the main contributions of the work in [46] is the evaluation of the listening quality for modern speech codecs, such as G.723.1, G.729, AMR and iLBC, to extend the applicability of the E Model. To this purpose, the PESQ-LQ index [47], an improvement of the PESQ index, is used to assess the quality of the speech with different codecs and at varying packet losses, modeled as a Bernoulli process. The results are then converted into the R factor and the resulting points are interpolated to achieve a three-parameter logarithmic expression of the impairment factor respect to the packet loss rate. These expressions are then used for playout buffer setting, minimizing the sum of delay and equipment impairment factors.

[48] proposes the use of simplified expressions of the equipment and delay impairment factors resulting from the study of Cole and Rosenbluth in [49]. Note that, the used simplifications are valid when the losses are random. The works proposed in [45, 46, 47, 48] are all based on the ITU-T E Model, which represents the most accurate tool for the required quality evaluation.

A detailed description of quality models will be presented in the next section.

2.3 Quality Model

In addition to the metrics already described previously, which define the characteristics of the network, there are additional metrics, which aim to quantify the performance experienced by the applications using the network. These metrics define the perception of application performance, experienced from the perspective of the end-users, which is also known as the user “quality of experience” (QoE). For IP-based voice and video applications the QoE is a compound metric dependent upon the quality of the encoder used, the quality of the service delivered by the IP network, and the quality of the decoder used. As such, QoE targets do not directly define the delay, jitter, loss etc., characteristics that a network should provide, but rather for a specified application, using a defined encoder/decoder, the network characteristics may be implied given a particular QoE target. QoE metrics can be measured by subjective or objective quality models.

Speech quality is determined by the listener’s perception, and hence it is inherently subjective. The Mean Opinion Score (MOS) [8] test is widely accepted as a norm for speech quality rating. However, the subjective MOS test is time-consuming and expensive. In recent years, several objective MOS measures were developed, such as Perceptual Evaluation of Speech Quality (PESQ) [50]. They measure the audible distortions based on the perceptual
domain representation of two signals, namely, a reference signal and a degraded signal which is
the output of the system under test. PESQ is based on MOS scale; however, that is subjective
model. This is the reason because it is prefer to use another objective quality model: the E
Model [30, 51].

ITU-T G.107 defines the E Model, a computational model combining all the impairment
parameters into a total value. The E Model is not a measurement tool, but an end-to-end
transmission planning tool; the output can be transformed into a MOS scale for prediction.

This section will describe in detail the characteristics of the 2 models currently used most
frequently: MOS and E Model.

2.3.1 Mean Opinion Score (MOS)

The most common speech quality metric is called the Mean Opinion Score (MOS). This
measurement scheme is described in ITU-T Recommendation P.800 [8]. Basically, MOS is a five-
point scale as follows:
- 5 = Excellent
- 4 = Good
- 3 = Fair
- 2 = Poor
- 1 = Bad

The objective with any coding technique is to achieve as high a ranking on this scale as
possible while keeping the bandwidth requirement relatively low. To determine how well a
particular coding technique ranks on this scale, a number of people (a minimum of about 30
people and possibly many more) listen to a selection of voice samples or participate in
conversations, with the speech being coded according to the technique to be evaluated. They
rank each of the samples or conversations according the five-point scale and a mean score is
calculated to give the MOS.

On the surface, this does not seem too scientific. Although the basis of the MOS test is a
matter of people listening to voice samples, there is more to it than that. ITU-T P.800 makes a
number of recommendations regarding the selection of participants, the test environment,
explanations to listeners, the analysis of results, and so on. For examples, when conducting MOS
test, there are certain phrases that are recommended to be used by the ITU-T. They are:
- You will have to be very quiet.
- There was nothing to be seen.
- They worshipped wooden idols.
- I want a minute with the inspector.
- Did he need any money?
Consequently, different MOS tests performed on the same coding algorithm tend to give roughly similar results. However, because the tests are subjective, variations occur. One test may yield a score of 4.1 for a particular coding algorithm, whereas another test may yield a score of 3.9 for the same algorithm. Therefore, one should treat MOS values with care, and at the risk of sounding cynical, MOS scores should perhaps be treated with some skepticism depending on who is claiming a particular MOS for particular coder/decoder (codec).

In general, and within design restrictions such as bandwidth, most coding schemes aim to achieve or approach toll quality. Although the definition of this term can vary, toll quality generally refers to an MOS of 4.0 or higher.

Each codec is given a MOS value based on any known impairments for the speed of the conversion, speech quality, and data loss characteristics. Below is a listing of the most common codecs used today for VoIP and their theoretical maximum MOS value (Table 2.3).

<table>
<thead>
<tr>
<th>Compression Method</th>
<th>Bit Rate (kbps)</th>
<th>MOS Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711 PCM</td>
<td>64</td>
<td>4.1</td>
</tr>
<tr>
<td>G.726 ADPCM</td>
<td>32</td>
<td>3.85</td>
</tr>
<tr>
<td>G.728 LD-CELP</td>
<td>16</td>
<td>3.61</td>
</tr>
<tr>
<td>G.729 CS-ACELP</td>
<td>8</td>
<td>3.92</td>
</tr>
<tr>
<td>G.729 x 2 Encodings</td>
<td>8</td>
<td>3.27</td>
</tr>
<tr>
<td>G.729 x 3 Encodings</td>
<td>8</td>
<td>2.68</td>
</tr>
<tr>
<td>G.729a CS-ACELP</td>
<td>8</td>
<td>3.7</td>
</tr>
<tr>
<td>G.723.1 MP-MLQ</td>
<td>6.3</td>
<td>3.9</td>
</tr>
<tr>
<td>G.723.1 ACELP</td>
<td>5.3</td>
<td>3.65</td>
</tr>
</tbody>
</table>

2.3.2 ITU-T E Model

Objective quality assessment methodologies of the conversational MOS make use of tools called opinion models [52]. These are aimed at evaluating the end-user perceived quality in two-way voice communication taking into account all the impairments introduced by the signal processing modules and network. The E Model is the opinion model initially developed and standardized by the ITU-T and subsequently also adopted by the ETSI, becoming the most widely used tool for objective quality assessment of the conversational quality. The main feature of the E-Model is the capacity of revealing the underlying causes of speech quality problems by means a set of relationships between an overall index, the R Factor, and metrics linked to: speech low bit rate coding; delay and loss distribution; frame erasure distribution; loss
concealment technique; architecture choices such as de-jitter buffer, packet and codec frame sizes.

The E Model is based on a mathematical algorithm, with which the individual transmission parameters are transformed into different individual "impairment factors" that are assumed to be additive on a psychological scale. The algorithm of the E Model also takes into account the combination effects for those impairments in the connection which occur simultaneously, as well as some masking effects. To the extent that impairments are present for which psychological additively is not maintained, E Model predictions may be inaccurate.

The relation between the different impairment factors and $R$ is given by the equation:

$$R = R_0 - I_s - I_d - I_{e,eff} + A$$

From left to right, the four terms are: the simultaneous impairment, delay impairment, equipment impairment, and the advantage factor, and they represent:

- $R_0$ = Basic Signal to Noise Ratio that represent subjective quality impairment due to circuit noise, room noise at sending and receiving sides, and subscriber line noise.
- $I_s$ = comprises the distortions introduced by the circuit-switched part of the transmission path and is frequently set to the default value of 6.8 [30]
- $I_d$ = Measures the impairments related to the mouth-to-ear delays encountered along the transmission path. Several values of this factor have been obtained with extensive experiments as a function of the average mouth-to-ear delay, which is given by the playout and encoding/decoding delay; by means of appropriate interpolation, it has then been possible to achieve an analytical expression [53].
- $I_{e,eff}$ = Represents the impairments associated with the signal distortion, caused by low bit rate codec and packet losses. It is mainly affected by the end-to-end packet loss rate and the speech codec used. The loss rate should comprise the packets that have been lost during transmission, after the application of the FEC loss recovery if used, and those that arrive correctly at the receiver but are too late to be played out. For a fixed loss rate, different impairment values are observed changing the number of frames inserted in a transport packet, the distribution of the packet losses, the sensitivity of the used codec to data frame losses, and the concealment algorithm. [54] provides the $I_{e,eff}$ values for some configurations.
- $A$ = Advantage Factor that it raises the level of conversational quality when the end-user may accept some decrease in quality for access advantage, such as mobility

Values of $R$ fall in the range of $0 < R < 100$, with higher values indicating higher speech quality. Table 2.4, taken from ITU-T Rec. G.109 [55], relates the E Model Ratings $R$ to categories.
of speech transmission quality and to user satisfaction. Note that connections with R-values below 50 are not recommended.

Table 2.4. Definition of categories of speech transmission quality

<table>
<thead>
<tr>
<th>R-value range</th>
<th>Speech transmission quality category</th>
<th>User satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 ≤ R &lt; 100</td>
<td>Best</td>
<td>Very satisfied</td>
</tr>
<tr>
<td>80 ≤ R &lt; 90</td>
<td>High</td>
<td>Satisfied</td>
</tr>
<tr>
<td>70 ≤ R &lt; 80</td>
<td>Medium</td>
<td>Some users dissatisfied</td>
</tr>
<tr>
<td>60 ≤ R &lt; 70</td>
<td>Low</td>
<td>Many users dissatisfied</td>
</tr>
<tr>
<td>50 ≤ R &lt; 60</td>
<td>Poor</td>
<td>Nearly all users dissatisfied</td>
</tr>
</tbody>
</table>

2.3.2.1 R factor working formula

The E Model (in its RFC) defines 20 different parameters each with a default value and their ranges of values are defined. If all parameters are set to the default values, the calculation results in a very high quality with a rating factor of \( R_{\text{default}} = 94.2 \), which is also defined as the intrinsic quality of a voice call with a mouth-to-ear delay of 0 ms. An adaptation made in the latest version of [30] reduces the value calculated standard parameters to 93.2: this small change (about 1%) on the value of R factor does detracts from the correctness.

For these reasons, the formula for R given above, can be simplified and in reality you use this working formula:

\[
R = 93.2 - I_d - I_e
\]

This simplified version of the formula of E Model allows us to focus better on the influence of delay and packet loss, the R factor. The two parameters that are remained, in fact, are direct function of the delay \( I_d \) and the loss of packets \( I_e \). The goal is obviously to find out how to maximize the value of R factor given the pair (delay, loss) as measured by the network. In a first analysis of the formula above stand out that the maximum value of R factor is obtained when \( I_d \) and \( I_e \) are both equal to 0. If it were true that \( I_d \) and \( I_e \) are released, or the delay affects the packet loss and vice versa, there would be nothing to discuss: should minimize the delay and try not to lose any packages.

Obviously, the hypothesis of independence of \( I_d \) and \( I_e \) is false precisely because the mechanisms that underlying the operation of the dejittering buffer, increasing the delay decreases the number of lost packets and vice versa. High is expectation that the packets
arrive at their destination, lower the probability of the presence of "holes" in the playout. Including the two variables so that "delay" and "loss" are inversely proportional, it will be possible with the appropriate approximations to calculate what is the point of best compromise that maximizes the value of R.

With the setting of dejittering buffer, the delay of packets changes and you get a certain loss. Now will discuss the approximations related to $I_d$ and $I_e$.

$I_d$. Also $I_d$, the impairment factor representing all impairments due to delay of voice signals is further subdivided into the three factors $I_{dte}$, $I_{dle}$ and $I_{dd}$:

$$I_d = I_{dte} + I_{dle} + I_{dd}$$

$I_{dte}$ gives an estimate for the impairments due to Talker Echo, $I_{dle}$ represents impairments due to Listener Echo and $I_{dd}$, a loss of interactivity, represents the impairment caused by too-long absolute delay $T_a$, which occurs even with perfect echo cancelling. These Impairments depends on 3 parameters:

- $T$: value of one-way delay ($I_{dte}$)
- $T_r$: value of the round-trip delay ($I_{dle}$)
- $T_a$: absolute value of the delay ($I_{dd}$)

The parameter $T$ refers to the one-way delay measured from sender terminal to receiver terminal.

The parameter $T_a$ was based on the mouth-to-ear delay that represents the sum of all delays between a terminal to another. Since it is assumed to put the terminal directly connected to the internet (without the help of other mechanisms for circuit switched call queue), we can say that the delays $T$ and $T_a$ are the same delay. The parameter $T_r$ but account for round-trip delay. Since we can assume that on average the delay is twice the delay $T$, mouth-to-ear, it can set equal to $T/2$. It follows that the approximations made so far lead to the equation:

$$T = T_a = T_r/2$$

Using these simplifications, $I_d$ can be considered only according to the delay $T$ and dependent only on the hypothetical systems do not put "circuit switched" between the two terminals in the call. So considering the interpolation of the values listed:

$$I_d = 0.024d + 0.11(d -177.3)*H(d-177.3)$$

Where $H(x)$ is the Heaviside step function and $d$ is the one-way delay ($H(x)=0$ if $x<0$; $H(x)=1$ otherwise).
The calculation of the relative function impairment due to the codec used is based only on interpolation of the data contained in [54]. The value of $I_e$, for a given codec and loss of the network, is a value that can be seen only experimentally. As demonstrated in [56], can approximate the series of numbers with the following fitting:

$$I_e = 11 + 40 \ln (1 + 10e)$$

The value $e$ refers to the percentage of loss.

With these simplifications, the working formula of R factor becomes:

$$R = 94.2 - 0.024d - 0.11(d_{177.3}H(d_{177.3})) - 11 - 40 \ln (1 + 10e)$$

The "d" and the "e" in the formula represent the approximate delay and loss of the communication, respectively. These variables can be decomposed into other variables directly correlated with the measurable parameters on the VoIP system, as following.

**d. Delay** can be decomposed thus:

$$d = d_{codec} + d_{dejitter} + d_{network}$$

Where:

- $d_{codec}$ is the delay introduced by encoding. This is the time required to sample and encode the audio before introducing it into an IP packet and send to receiver. Assuming you have a codec with the following features, create packages of 10 ms using a look-ahead of 5 ms. This means that it is $d_{codec}$ output:

$$d_{codec} = N \times 10 + 5$$

Where $N$ is the number of codec packets contained in the IP packet and the delay is expressed in ms. In this case, if there are 2 audio packets into an IP packet $d_{codec}=25$ms. This delay is a requirement of the VoIP system and it's impossible to change this delay once chosen and configured the codec used for VoIP transmission.

- $d_{network}$ represents the time when the packet travels from sender to receiver into the IP network. The causes of this delay are several, for example different routes, delays created by queues into switches or routers in the network and so on. The $d_{network}$ is the variable delay that seeks to reduce, by inserting the dejittering buffer: the ultimate goal of the operation is to stabilize the delay to the receiver so you can successfully run the periodic playout of packets. Obviously this is a requirement of the network on which it is impossible to act; can only measure it and use it to modify the dejittering buffer.
- $d_{dejitter}$ represents the delay added by the dejittering buffer. This is the only delay that it's possible to control: using other available data to search a value that enables it to achieve an acceptable loss and total delay (the sum of the 3 delays) which is as low as possible.

**e.** Loss can be decomposed thus:

$$e = e_{\text{network}} + (1-e_{\text{network}})e_{dejitter}$$

Where:

- $e_{\text{network}}$ represents the loss factor due to the network: is the percentage of packets lost (which not arrive at destination) than those sent by the sender. On the Internet IP packets are characterized by the TTL (time to live): This is the number of steps through entity of ISO-OSI level 3 that a packet can do before being discarded and declared lost. If the packets meet congestion at some point in the network and/or perform too many steps before reaching destination, they are then discarded. The $e_{\text{network}}$ is requirement that comes from the network: it’s possible to measure it at any time considering how many packets did not reach its destination. It’s impossible to reduce it in any way; it is a feature of network and the variations depending on the load of equipments in the time in which it is measured.

- $e_{dejitter}$ represents the loss due to dejittering buffer, it is a consequence of the length of choice $d_{dejitter}$: less additional delay is introduced, more packets are discarded due to a time of playout premature. For this reason $d_{dejitter}$ and $e_{dejitter}$ are linked together.

### 2.3.3 Correlation R factor-MOS

Correlations between R factor and MOS scores and the user's experience of VoIP call quality are shown in the following table (Table 2.5):
Table 2.5. Correlation between R factor and MOS

<table>
<thead>
<tr>
<th>R-Factor</th>
<th>Quality of voice rating</th>
<th>MOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 &lt; R &lt; 100</td>
<td>Best</td>
<td>4.34 - 4.5</td>
</tr>
<tr>
<td>80 &lt; R &lt; 90</td>
<td>High</td>
<td>4.03 - 4.34</td>
</tr>
<tr>
<td>70 &lt; R &lt; 80</td>
<td>Medium</td>
<td>3.60 - 4.03</td>
</tr>
<tr>
<td>60 &lt; R &lt; 70</td>
<td>Low</td>
<td>3.10 - 3.60</td>
</tr>
<tr>
<td>50 &lt; R &lt; 60</td>
<td>Poor</td>
<td>2.58 - 3.10</td>
</tr>
</tbody>
</table>

The MOS is related to R factor by the following equations:

\[
\text{For } 0 < R < 100 \quad MOS = 1 + 0.035 \cdot R + 7 \cdot 10^{-6} \cdot R \cdot (R - 60) \cdot (100 - R)
\]

\[
\text{For } R < 0 \quad \text{MOS}=1
\]

\[
\text{For } R > 100 \quad \text{MOS}=4.5
\]

2.4 E Model-based quality maximization algorithm

Quality-based category of algorithms is driven by the maximization of a metric linked to the end-user perceived quality, as mentioned. These algorithms appeared in the literature only a few years ago, and have been triggered by the advancements reached by the standardization committees working on the development of voice QoS evaluation tools within ITU, ETSI, and ANSI organizations. Now, I have presented all issues that affect the QoS in a VoIP service and the 2 most used quality model (MOS and E Model) to quantize a VoIP conversional quality.

The setting of the size of the buffer is a key operation for it directly affects the packet loss rate and the conversational interactivity, which quite influence the quality perceived by the end-user. Buffer adjustments may be introduced to have the optimal balance between loss and delay on the basis of a reference quality model.

The goal is to combine a playout buffer with this quality model, in a manner that playout delay is continuously updated in every silence period as in an intra-talkspurt approach. This is possible and this approach appears easy, but in practice, the implementation is not very easy because networks have a lot of variables that we must consider.

The initial point is the choice of the quality model. Nowadays E Model represents the best objective quality model because it considers directly the main problems of quality degradation as delay and loss.
The techniques examined hitherto are aimed at setting the playout delay so that the fraction of expected late packets is either negligible or lower than a desired threshold. To fulfill such an objective, these approaches may introduce high end-to-end delays under certain network conditions, with potential significant degradations of the conversational quality. To prevent this drawback, the setting of the loss rate threshold should also take into account the network status. A straightforward solution is to rely on the quality perceived by the end-user and to maximize it through a correct balancing of the delay versus the loss. It requires to analyze the importance of the loss and delay impairments affecting the conversational quality and to find the optimal compromise according to an opinion model.

This solution is sketched in Figure 2.11. The network delay is observed for every packet and used to update an estimation of the delay distribution $F_D(d)$, usually making use of a parametric estimation procedure. Losses during transmissions are also taken into account by regarding the lost packets as having undergone an infinite delay. During the silence period (with the obviously utilization of Voice Activity Detection technique linked to the codec), this distribution is used to compute the packet loss rate $L(P_D)$ function. Now there are the two metrics to calculate the R factor: delay distribution $F_D(d)$ and packet loss rate $L(P_D)$ function.

![Figure 2.11. Sketch of the quality-based playout approach](image-url)
First metric depends on the packet delays $d$ that arrive at the receiver side; second metric depends on the choice of playout delay of the buffer $P_D$. This value of $L(P_D)$ is introduced in the working formula of E Model, obtaining an $Q(P_D)$. After, if $P_D$ varies in a little range of value (few milliseconds), it’s possible to obtain several $Q(P_D)$, one for each $P_D$ values into this range. $Q(P_D)$ represents just the $R$ factor dependent on $P_D$. Then, the $P_D$ that maximize the $R$ factor is the optimal playout delay $P_D^*$ of the quality-based dejittering buffer, and it is used to setting the next talkspurt.

This dejittering buffer has needed only an initial setting: the $P_d$, but only for the initial conversation. Then algorithm calculates all of its parameters at runtime, it is an interactive process. The value of playout delay $P_d$ varies automatically with every talkspurt maximizing the objective quality through analysis of delay and loss.

This represents the initial quality-based approach, which in the next chapters will see its specific application on NGNs.
Chapter 3 – Control of the Playout Delay over a Satellite Network

In streaming applications, playout buffering is operated at the receiver side to compensate network delay variations. This function allows the receiver to reduce packet lateness at the expenses of an increase in the end-to-end delay. IP Telephony is one of the applications where this function is crucial since transmission delays heavily affect the interactivity and significant losses injure the signal quality. Many approaches have been proposed and analyzed in the previous chapters to control the playout delay in IP Telephony, where the most recent presented buffer rely on the use of the ITU-T E Model to evaluate the quality perceived by the end-user and to maximize the perceived quality through the control of playout delay.

However, none activity research of these specifically deals with the scenario where the service is provided over satellite networks. Indeed, nowadays, this scenario is quite frequent with the reduction of the costs of the satellite channels and equipments. In this case, there are some specific issues that need to be dealt with to improve the effectiveness of the playout control algorithms. These issues are related to the delays that are encountered when satellite links are involved. These aspects are described in this chapter, which also presents a new quality-based playout algorithm that provides better results with respect to alternative approached when satellite transmissions are involved.

The chapter is organized as follows. Initially describes the GEO satellite network, the GEO speech frame model and the delays that are observed in the end-to-end communication. Starting from this analysis, a novel ad-hoc playout mechanism is proposed and we present the analysis of the impact on the end-to-end data transport performance of the satellite links. Finally, comparative results are provided and conclusions are drawn.

3.1 The CNIT GEO Satellite network

In the last years, the satellite system has become a suitable alternative to the conventional fixed/terrestrial backbone, being:

i) distance-insensitive both for the QoS (delivery at almost identical quality of service across an endless number of locations) and hardware connection (theoretical capability of reaching all the sites in the network regardless of their distance);
ii) flexible, since many satellite IP networking equipment are available to help in covering long-distance connections with easy to configure procedures.

Thanks to these advantages, there have been many efforts towards the integration of satellite channels with terrestrial networks making also use of QoS technologies to assure the required QoS [57, 58, 59].

In 2000, the CNIT (Italian University Consortium of Telecommunications) has set up its GEO satellite network to create a corporate network connecting its research laboratories at the main Italian Universities (Figure 3.1). This solution allows the members of the consortium to share both common Internet services and audio/video/data, with very high quality and reliability. The core of this satellite network is the Skyplex technology [60] that implements an IP based satellite network with mesh, star or hybrid topologies.

Skyplex is a digital multiplexer, based on the European standard DVB (Digital Video Broadcasting) and compatible with the DVB/MPEG and the DVB-RCS technologies. The network operates in the Ka-band (20-30 GHz) and provides a guaranteed bandwidth of 2 Mbps shared among the connected terminals. In the typical applications, most of this bandwidth is reserved for audio and video multicasting, so that a significant improvement of their quality is achieved. Furthermore, the bandwidth requirements of each terminal are periodically adjusted by means of a BoD (Bandwidth on Demand) mechanism, following the indications of a Regional Network Control Centre (RNCC).

Figure 3.1. CNIT – Skyplex Satellite network
EUTELSAT [61] is the space operator partner of the network. The satellite Hot Bird 6 (positioned at 15 degrees East) covers all European countries through 4 uplink areas, one for each up-link spot-beam. These regions also correspond to 4 geographical areas covering Italy, Germany, France and Great Britain, and Spain.

Hot Bird 6 has 8 Skyplex units on board, with following characteristics:
- 8 Skyplex units on board;
- 18 x 2 Mbps channels that operate both in continuous mode (SCPC) and in burst mode (TDMA) [62, 63];
- TDMA is divided into 6 users;
- 350 kbps - 2 Mbps TDMA for each user depending on the traffic pattern.

Tests have been conducted in a burst mode with timeslots assigned in a best effort mode. Experiments are mainly related to the two communications between two sites, i.e., Cagliari (UniCA) and Florence (UniFI), by means of the Skyplex technology station with an Ethernet interface. Every site implements an IP Telephony solution based on Asterisk, an open source PABX-IP (Private Automatic Branch eXchange-IP) [64], implementing the IETF SIP protocol.

The Skyplex On-Board Processor (OBP) originates from the idea of moving the MPEG2 Transport Stream (TSs) multiplexing capability to the satellite. Multiplexing is usually performed within the ground stations for TV broadcasting services. In Skyplex, the audio/video content is transmitted to the satellite with proprietary Time Division Multiple Access (TDMA) or Single Channel Per Carrier (SCPC) up-links by sparse and relatively small Earth Stations (E/Ss). These signals are demodulated, regenerated and multiplexed by the Skyplex units on-board the satellite and forwarded in the down-link in standard DVB-S format. Application of this technology for data connectivity imposes a need for the ground stations to both receive and transmit, so permitting the implementation of a network in which all terminals can communicate with each other over bi-directional links.

On the one hand, the use of Ka band permits to increase the antennas gain (or G/T ratio) and EIRP of the satellite. On the other hand, the introduction of turbo encoding has the further advantage of smaller E/Ss in terms of antenna size and amplifier power that are within the limits for the individual licensing exemption. The direct consequences are the reduction of equipment cost maintaining a good performance and reliability level.

The Skyplex Data system is based on some DVB-RCS features. The turbo encoding, compliant with the standard, contributes to a reduction in the size and cost of the ODU (antenna and amplifier). On the other hand, the Multi-Protocol Encapsulation (MPE) [65], i.e. the insertion of data into the MPEG2 Transport Stream, supplies the means to transmit the IP packets to the satellite. Here, the up-link data is multiplexed and forwarded in a single stream on the downlink to be received in IP/DVB stream format (Figure 3.2).

The E/Ss of the Skyplex Data network clearly belong to the Very Small Aperture Terminals (VSATs) category. The terminals are composed of an indoor unit (IDU) equipped with the Skyplex proprietary modulator, the DVB demodulator and components for the encapsulation
(transmission) and extraction (reception) of the IP traffic from the MPEG packets. All are integrated in a stand-alone box. The external interfaces are the Ethernet network interface, the IF connectors, a console port for the configuration and management of the terminal, and an ASI interface for the foreseen future development of TV services.

Figure 3.2. Data processing in a Skyplex transmission

In a standard configuration of the terminals, the outdoor unit (ODU) consists of:
- a 90cm antenna
- a transceiver composed of a Ka band transmitter (block up-converter followed by a 2W SSPA) and an LNB, again in Ka band (20MHz). The IF interfaces are in S band for transmission and L band for reception.

This equipment guarantees an up-link rain-fade margin of about 6dB in the 15 dB/K coverage, ensuring link availability for 99.8% of the time. Furthermore, such antennas are compliant with the rules for exemption of individual licensing.

The control and management of the network follows a hierarchical structure that reflects the organization of the entire network itself. On the top, there is the Network Control Center (NCC) which has a ‘global’ network management role and communicates via a terrestrial link with the Regional NCCs (RNCCs). Each RNCC controls a subset of the network defined as a region: such subdivision is for synchronization purposes. The RNCCs generate the messages sent to the Traffic Terminals (TTs) used for acquisition and synchronization, and compute the BTP for the assignment of the capacity (in terms of time slots or bursts) to the terminals. The BTP is calculated from the CIR programmed and on the basis of the bandwidth request messages from the TTs. Signaling messages are in IP format and use two dedicated PIDs per region: the first is for the multicast delivery of the messages from the RNCC to the terminals and the second is shared among the TTs to transmit bandwidth requests to the RNCC. Signaling messages are transmitted in a burst allocated and assigned by default to every terminal in the network. This ‘signaling burst’ may also carry user: the portion of the burst not used for signaling can be used for normal traffic transmission.

The RNCCs can communicate with the entire satellite network using the Gateway Terminals (GTs). These do not physically differ from the TTs but have different roles inside the
network. In particular, the main function of GTs connected within the RNCCs’ VLAN is to transmit and receive the management messages. Finally, the regions are sub-divided into populations grouping together terminals with similar properties, i.e. the same type of ODU (because of reference frequencies to be used) belonging to the same customer, same type of services required, etc. Within their region, the GTs belong to a population created by default. The network developed has the NCC, RNCCs and GTs installed in Telespazio’s station in Lario. This is an area that is well covered by three up-link spot beams (respectively covering Italy, France and Germany) and, albeit with a lower value of G/T, by the secondary side-lobe of the Spanish coverage. From this position, it is thus possible to reach all the Ka band coverage of the satellite and to control the complete network, installing all the RNCCs in the same location.

The Skyplex Data network is a new answer to requests for connectivity for broadband multimedia applications. It supports high-rate point-to-point communications based on IP with reliable and unreliable transport protocols, and the tests demonstrate that it is particularly attractive for real-time interactive applications. Furthermore, the terminals can be configured to transmit in multicast mode exploiting in the most natural way the characteristics of the TDM downlink of the system. In the remainder of this paper, this aspect is discussed in depth. In the unicast transmission, the well-known constraints in terms of performance of TCP/IP via satellite could be compensated for by using a TCP “acceleration system” similar to that used in the Linkstar system [66]. Such a method for acceleration of the TCP data transfer was briefly tested also with Skyplex, but it is not included, for the moment, in the set of features available with this system. To improve the system, it is also planned to add to these principal features the possibility to broadcast audio and video in MPEG2 format. The terminals are, in fact, already equipped with an ASI interface, which will be used to insert the contents to be transmitted within the framework of a ‘micro-TV’ service.

3.2 Transmission delays over Satellite Networks

The Skyplex TDMA structure [63] is organized in two different levels: user timeslots and MPEG2 burst. At a first level of analysis, every TDMA frame is made of $N$ user timeslots for each user connected to the GEO satellite network. The user timeslot contains $M$ MPEG2 transport packets. The detailed description regarding the process of creating the MPEG2 transport stream from the speech codec packets is provided in [63].

The Skyplex TDMA uplink frame organization is shown in Figure 3.3. The uplink TDMA frame is constant for all modes and is uniquely defined as the equivalent time for the on-board multiplexer to transmit 48 consecutive DVB packets from a given Skyplex channel. (Note that, a DVB packet is 204 bytes in length.)
As shown in Figure 3.3, the equivalent time of 4 DVB packets is allocated for guard time and others synchronized data. The overhead ratio of 52/48 (or 8% of overhead) is constant. The burst length depends on the number of active TDMA stations.

Hence, in the Skyplex TDMA structure $N$ and $M$ are combined in a way that $N \times M = 48$. This is the default configuration, as suggested by the network provider for optimizing the bandwidth usage versus the number of connected users. Furthermore, the resources of the network are assigned to the users with the already cited BoD mechanism. That is, first of all the RNCC monitors the network and then produces a ticket, which is sent to all the satellite gateway terminals to which the users are directly connected every interval of 820 ms. This ticket contains the exact satellite backbone capacity assigned by the network control centre to all the users on the basis of their requirements.

As a result of this frame structure, a variable number of MPEG2 packets per TDMA frame are assigned to each user. In fact, the first user timeslot contains, other than the MPEG2 transport stream packets, the ticket produced by the RNCC about the network capacity given to that user. Instead, the next frames, of duration of 820 ms, include several RTP packets, because there is no need for additional traffic information. This structure containing few/many MPEG2 packets is repeated at fixed intervals for the entire duration of transmission.

This analysis is of crucial importance for the study of the end-to-end delays encountered by the RTP packets. From what has been said, the Skyplex proprietary structure of the frame modifies the frame stream: all the RTP packet delays are dependent on the last RTP packet of the TDMA frame. In fact, the transmission over the satellite link can start only when the last RTP packet closes the frame. As a consequence, the associated end-to-end delays of the RTP packets present the behavior showed in Figure 3.4, which is a typical “saw tooth” trace.
The control of the playout delay needs to take into account this “saw tooth” effect not to introduce too high delays. Accordingly, in the following I present the analysis of the delay components that characterize the considered architecture.

At both sites, the transmission through the segments of IP networks introduces two delay components, which we refer to with $D_1$ and $D_2$, respectively. These are random delays that follow the temporal variability of the traffic loads and congestion occurrences. Two constant delays are then added, which are the transmission delay $t_x$ of a generic RTP packet through the satellite link and the propagation delay $t_{prop}$ of the electromagnetic wave. Both these quantities have to be considered twice for the go and back links. $t_{prop}$ is equal to $(\text{sat distance})/(\text{speed})$ (NOTE: $= (35790\text{ km})/(300\text{ km/s})$). The last component is $D_3$: it is the delay introduced to each packet transmission that waits at the satellite transmitter buffer till the satellite frame is full and ready to be transmitted. Based on this analysis, the end-to-end delay for a generic RTP packet is the following:

$$d = D_1 + D_2 + D_3 + 2t_x + 2t_{prop}.$$ (1)

We now introduce a subscript to index the delay according to the position of each packet in a satellite frame; packet 0 is the last in frame, packet 1 is the second last, and so on, so that the delays can be written as follows:
\[ d_0 = D_0^i + D_0 + 2t_x + 2t_{prop} \]
\[ d_1 = D_0^i + D_1^2 + 2t_x + 2t_{prop} + D^{int} \]
\[ \vdots \]
\[ d_i = D_0^i + D_i^2 + 2t_x + 2t_{prop} + i \cdot D^{int} \]

Note that since the frame is transmitted when the last packet arrives at the satellite transmitter, the delay of every packet includes \( D_0^i \). In (2), \( D^{int} \) is the length of the speech frames transmitted in a RTP packet and depends on the specific speech codec adopted in the VoIP system.

In this way, we have been able to remove the random component \( D_x \) in (1) and the delay is then influenced by the random components \( x = D_1^1 + D_2 \). Predicting the probability density function (pdf) \( f(x) \) of \( x \), we are able to obtain the pdf for all the packets \( f_{d_i}(d_i) \) in a frame according to their position within the frame:

\[ f_{d_0}(d_0) = f(x - 2t_x - 2t_{prop}) \]
\[ f_{d_1}(d_1) = f(x - 2t_x - 2t_{prop} - D^{int}) \]
\[ \vdots \]
\[ f_{d_i}(d_i) = f(x - 2t_x - 2t_{prop} - i \cdot D^{int}) \]

### 3.3 Playout Buffering

The playout buffering algorithm proposed relies on the quality-based approach that is presented in the previously chapter, that is, it is designed to adjust the buffer size so as to maximize the expected end-user perceived quality. The aim is to jointly consider the expected end-to-end delay and information loss making use of a perceptually motivated optimality criterion that allows the receiver to automatically balance packet delay versus loss. In this way, no thresholds for the loss and/or delay have to be considered: the playout algorithm automatically adapts the buffer size so as to maximize the expected quality. The following main operations are performed: statistics relevant to loss and delay are predicted by means of the previously sent packets; based on this information, the buffer setting is accomplished so as to maximize the expected conversational quality during future conversational units. To evaluate the quality, as mentioned, it makes use of the ITU-T E Model. Accordingly, the overall output index is a function of several system parameters, including the dejitter buffer size, that it’s
summarized by a working formula. I briefly summarize the expression of the R factor which represents the output index of the E Model that measures the user-perceived quality:

\[ R = 93.2 - I_d - I_e \]

Where \( I_d \) and \( I_e \) depend on playout delay (the total end-to-end delay) and packet loss, respectively. The loss and end-to-end delay can be expressed in terms of components relevant to the codec, network and playout buffer in the end-to-end path as follows:

\[
\begin{align*}
E &= c_{net} + (1 - c_{net})e_{dejitter} \\
P_D &= d_{codec} + d_{dejitter} + d_{net}
\end{align*}
\]

The total network delay \( d_{net} \) is the same delay \( d \) used in the previous section. The codec delay is a constant value which is often neglected. The R factor is then a function of the total packet loss rate and the end-to-end delay, named also playout delay. Note that the amount of losses depend on the playout delay too: the higher the delay is, the lower the loss rate is.

Finding the optimal playout delay for a subsequent talkspurt during the silence period according to the E Model is the aim of the proposed strategy, whose main steps are shown in Figure 3.5.

Figure 3.5. Sketch of the proposed playout control strategy

65
At the reception of each packet, the observed delay \( d \) is used to update the pdf \( f(x) \) defined in the previous section. Additionally, the length of each received TDMA frame in terms of number of RTP packets conveyed is used to build the probability function \( p(s_i) \): it is a function representing the probability of transmitting a packet in position \( i \) within a frame. \( f(x) \) and \( p(s_i) \) during each silence period are then used to represent the relationship between late packet loss probability and playout delay:

\[
e_{\text{dejitter}} = L(P_D) = \sum_{i=0}^{\max} p(s_i) \cdot \int_{d_i}^{\infty} f(d) \, dx = \sum_{i=0}^{\max} p(s_i) \cdot \int_{P_D - 2t_i - 2P_{\text{min}} - tD^m}^{\infty} f(x) \, dx
\]

where max represents the maximum number of RTP packets that can be conveyed in a TDMA frame. The following two steps in the proposed algorithm are quite easy to follow. Function \( L(P_D) \) is introduced in the expression of \( E \) together with the average network packet loss ratio (\( e_{\text{rec}} \)). In this way the final expression of R factor as a function of only \( P_D \) is available. The following operation consists in to finding the optimal \( P_D^* \) value that will be used during next talkspurt, that is, the one that maximize R factor.

To increase the effectiveness of the proposed playout control scheme, an essential condition is to define a simple statistical method for estimating \( f(x) \) on the basis of observed measurements. Indeed, network characteristics vary with time and, consequently, the delay statistical trends do; a continuous update of this information is then required to take into account these evolutions. Several approaches can be used to predict statistical trends of network delay relying on the analysis of historical and current information. The full aggregation method accumulates data into the probability distribution curve, giving the same weight to old and recent samples; this approach makes the system unable to quickly react to network traffic variations. Differently, the flush and refresh method builds the probability distribution curve on the basis of the last observed \( M \) packets and leaves out historic information, generating high overhead. An intermediate approach that we have decided to adopt is the store and track method. This approach does not discard the old data entirely, but gradually reduces its effect on the histogram to approximate the statistical distribution. Each value in the histogram counts the number of occurrences of packets sent within a certain interval of delays (bin). Note that the shorter the bin is, the higher the accuracy in the estimation of \( f(x) \) is. On the other hand, short bins increase the algorithm complexity when maximizing the R factor. To reduce the weight of older samples, each bin of the histogram is periodically scaled down by an aging factor \( F \).

As to the maximization of the R factor, it is a procedure conducted numerically: the packet loss rate is evaluated function at several \( P_D \) values according to the temporal resolution; this
are than used in the linear expression of the R factor; in the current implementation the optimal $P_D$ value is found looking at all the computed R factor values.

3.4 Measurements and Results

Conducting experiments to test a particular solution requires setting up a testing environment suitable for the purpose, it is necessary to define briefly the fitting hardware and software.

First of all we have used an IP PaBX (Private automatic Branch eXchange over IP network) to connect the two sites (Cagliari and Florence). The IP PaBX used is Asterisk [64], that is an open source software that turns an ordinary computer into a voice communications server produced by Digium. In reality, the asterisk was two servers, one for each site. Connected to each Asterisk there is a laptop with softphone (software that reproduces an IP Phone) installed in order for the voice communication. Then we used a network protocol analyzer based on network data capturing technology: Wireshark [67]. It allows examining data from live satellite network. Matlab [68] is the software that was used to analyze the data collected by recording the track and emulate the operation of various types of buffers.

I have recorded and evaluated several traces at different traffic loads of the GEO satellite network. An overview of the main features of these traces is provided in Table 3.1. All the tests have been performed synchronizing both sides clocks by means of the NTP (Network Time Protocol).

<table>
<thead>
<tr>
<th>Codecid</th>
<th>GSM, iLBC, G.711 µ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call duration (min)</td>
<td>5-10</td>
</tr>
<tr>
<td>Average network delay (sec)</td>
<td>0.354-0.400</td>
</tr>
<tr>
<td>Network delay variance (sec)</td>
<td>0.0034-0.0040</td>
</tr>
</tbody>
</table>

The “saw tooth” effect is clearly independent from the used codec and a single tooth can be seen as a network spike (sudden onset of a large increase in network delay, followed by a series of packets arriving almost simultaneously, leading to the completion of the spike). For this reason, we have compared the performance of this strategy (named Ad-Hoc Sat) with the class of spike detection playout control algorithms. A delay spike can be detected when the difference in delay values for the two most recently received packets is greater than a certain
threshold. Briefly, in these algorithms there are two modes, normal mode and spike detection mode, and a threshold, which depends on the buffer used, separates the choice of two modes. In normal mode, it operates in one way, but in spike detection mode, the playout delay is updated differently.

Specifically, the Adaptive Linear Filter Buffer with spike detection [69] and the Enhanced-Normalized Least Mean Square (E-NLMS) [70] have been chosen. Among the adaptive playout techniques, the first strategy is one of the most used because it offers good performance and a straightforward implementation. The E-NLMS offers results comparable to the Linear Filter. The strategies have been compared in terms of the R factor over a number of random talkspurts. In this chapter of PhD thesis I present the results of such comparison for one significant and representative trace: a 5 minutes conversation with GSM codec. The RTP packet delays for the considered trace are presented in Figure 3.6, while the resulting R factor for the initial talkspurts is shown in Figure 3.7.

Figure 3.6 shows the typical “saw tooth” delay trace, as it was presented and commented previously. Figure 3.7 provides a comparison of the three techniques for the first 30 talkspurts. The initial value is 0 because the initial playout delay is set to 100 ms for any algorithms. We can observe as the performances of Ad-Hoc Sat are constant and the value of R factor remains around 83, which is a high-quality level of voice rating. The reason is that the optimal value of the playout delay is calculated for each talkspurt considering both the R factor maximization and the inner structure of the TDMA frame, that is, the position of the RTP packets with their probability. The Linear Filter results are affected by continuous R factor changes from packet to packet.

This highlights that this technique is not able to take into account high spikes of delays between two close RTP packets, resulting occasionally in burst of losses and audible quality
degradation. This feature is improved by the E-NLMS because of an additional parameter that allows adjusting the buffer better when high delay variations occur.

![Graph showing comparison of alternative techniques in terms of R factor.](image)

Figure 3.7. Comparison of the alternative techniques in terms of R factor.

Table 3.2 show the average R factor values excluding the first initial five talkspurts. It highlights that the proposed algorithm performs more than 4 points better than the Linear Filter and almost 1 point than the E-NLMS in terms of R factor. If we take a look at the delay, we see that the E-NLMS is able to obtain good results at the expense of delays that are higher than the Ad-Hoc Sat ones due to the oversetting of the delay when spikes are detected.

Table 3.2. Average R factor and playout delay

<table>
<thead>
<tr>
<th></th>
<th>Linear Filter</th>
<th>E-NLMS</th>
<th>Ad-Hoc Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean R factor</td>
<td>78.7</td>
<td>82.3</td>
<td>83.2</td>
</tr>
<tr>
<td>Mean Playout Delay (sec)</td>
<td>0.485</td>
<td>0.5155</td>
<td>0.498</td>
</tr>
</tbody>
</table>

3.4.1 An overview of Adaptive Linear Filter Buffer

An approach to dealing with the unknown nature of the delay distribution is to estimate these delays and adaptively respond to their change by dynamically adjusting the playout delay.
by a linear recursive filter characterized by the weighting factor $\alpha$ ($\alpha = 0.998002$). Specifically, the delay estimate for packet $i$ is computed as:

$$\hat{d}_i = \alpha \cdot \hat{d}_{i-1} + (1 - \alpha) \cdot n_i$$

and the variation is computed as

$$\hat{v}_i = \alpha \cdot \hat{v}_{i-1} + (1 - \alpha) |\hat{d}_i - n_i|$$

Where

- $t_i$: the time at which packet $i$ is generated at the sending host
- $a_i$: the time at which packet $i$ is received at the receiving host
- $p_i$: the time at which packet $i$ is played out at the receiving host
- $v_i$: the queuing delay experienced by packet $i$ as it is sent from the source to the destination host
- $b_i$: the amount of time that packet $i$ spends in the buffer at the receiver, awaiting its scheduled playout time, $b_i = p_i - a_i$
- $d_i$: the amount of time from when the $i^{th}$ packet is generated by the source until it is played out at the destination host, $d_i = p_i - t_i$ (this will be referred to as the “playout delay” of packet $i$)
- $n_i$: the total “delay” introduced by the network, $n_i = a_i - t_i$. Because it have not assumed that the sender and receiver clocks are synchronized, $n_i$ may not be equal to the actual delay experienced by the packet.

The playout buffer used in my PhD research is an Adaptive Linear Filter Buffer with spike detection. When submitting a spike, a threshold detects and changes in spike mode, which is different to uploading the delay statistics than normal mode. Detection of the beginning of a spike is very simple:

$$\text{if } \text{abs}(n_i - n_{i-1}) > \text{spike threshold} \text{ then } \text{mode} = \text{IMPULSE};$$

Once that is in impulse mode on detection of a spike, it seems natural for us to “follow” the spike. Thus, in impulse mode, it is allowed our estimate to be dictated only by the most recently observed delay values. Specifically,

$$\text{if } \text{mode} = \text{IMPULSE} \text{ then } \hat{d}_i = \hat{d}_{i-1} + n_i - n_{i-1};$$
The detection of the completion of a spike is a bit tricky because in certain cases the delay on completion of the spike can be different from the delay before the beginning of the spike. Nonetheless, one prominent characteristic was that a series of packets would arrive one after another almost simultaneously at the receiver, and almost immediately following the observed increase (upward spike) in delay. Since the packets within a talkspurt are transmitted at regular intervals at the sender, near simultaneous arrivals implies that subsequent packets in the burst of arrivals have experienced progressively smaller end-to-end network delays. For this reason there is a variable \( \text{var} \) with an exponentially decaying value that adjusts to the slope of spike. When this variable has a small enough value, indicating that there is no longer a significant slope, the algorithm reverts back to normal mode. As shown in the pseudo code below, \( \text{var} \) uses the two most recent delay observations together with the current value, \( n_i \), to track the “slope” of the spike. In the case of a spike, abs value is going to be non-zero and will determine whether the spike has ended or not.

1. \( n_i = \text{Receiver timestamp} - \text{Sender timestamp} \)
2. if \( (\text{mode} == \text{NORMAL}) \) {
   if \( \left| \text{abs}(n_i - n_{i-1}) - \text{abs}(\hat{d}_i) \right| > 2 + 800 \) {
     \( \text{var} = 0 \); /* Detected beginning of spike */
     \( \text{mode} = \text{IMPULSE} \);
   }
   else {
     \( \text{var} = \text{var}/2 + \text{abs}((2n_i - n_{i-1} - n_{i-2})/8); \)
     if \( \text{var} <= 0.03 \) {
       \( \text{mode} = \text{NORMAL}; /* End of spike */ \)
       \( n_{i-2} = n_{i-1}; \)
       \( n_{i-1} = n_i; \)
       return;
     }
   }
}
3. if \( (\text{mode} == \text{NORMAL}) \)
   \( \hat{d}_i = 0.125 + n_i + 0.875 \times \hat{d}_{i-1}; \)
else
   \( \hat{d}_i = \hat{d}_{i-1} + n_i - n_{i-1}; \)
   \( \hat{v}_i = 0.125 \times \text{abs}(n_i - \hat{d}_i) + 0.875 \times \hat{v}_{i-1}; \)
4. \( n_{i-2} = n_{i-1}; \)
   \( n_{i-1} = n_i; \)
return;

### 3.4.2 An overview of Enhanced-Normalized Least Mean Square (E-NLMS)

The E-NLMS algorithm improves the basic NLMS predictor by adding a spike-detection mode to the algorithm. During normal mode of operation, the E-NLMS algorithm works exactly
like the basic NLMS predictor. A delay spike is detected when either the previous packet was late (it arrived after it was supposed to be played out) or the actual network delay exceeds the predicted delay value by a threshold.

The variation in the network delay, $\hat{v}_i$, is equal to previous Linear Filter:

$$\hat{v}_i = \alpha \hat{v}_{i-1} + (1 - \alpha) |\hat{d}_i - n_i|.$$  

While, total end-to-end delay, $D_i$ is then calculated as:

$$D_i = \hat{d}_i + \beta \hat{v}_i$$

Where $\beta$ is a safety factor used to moderate the tradeoff between end-to-end delay and packet ‘loss’ rate due to late packets. The $\beta \hat{v}_i$ term is a safety buffer term used to ensure that the end-to-end delay is large enough, so that only a small portion of received packets will arrive too late to be played out. The value of $\alpha$ and $\beta$ are set to 0.998002 and 4.0, respectively.

Upon switching to spike mode, the playout delay is still based on the NLMS delay prediction. However, as the NLMS algorithm overestimates the network delay for packets following the beginning of a delay spike, the value of the safety factor, $\beta$, can be decreased, thereby reducing both the safety buffer term, $\beta \hat{v}_i$, and the end-to-end delay, $D_i$, during the spike. The end of the spike is detected and the mode of operation switches back to normal mode when the NLMS delay prediction, $\hat{d}_i$, no longer exceeds the actual network delay, $n_i$.

As a spike is characterized by a sudden, large increase in network delay followed by declining network delays, the bi-modal algorithm takes advantage of this trait by reducing the safety buffer term, and thus tracks the network delay more accurately. To avoid having the playout delay fall too low, the end-to-end delay, $D_i$, is not allowed to fall below the delay estimate.

The pseudo code of the algorithm is given here:
\[ d_i = w_i^T n_i; \]
\[ w_{i+1} = w_i + \mu/(n_i^T n_i + \alpha) n_i c_i; \]

if ( mode == SPIKE )
    \[ ARdelay_i = \alpha ARdelay_{i-1} + (1 - \alpha) n_{i-1}; \]
    \[ v_i = \alpha v_{i-1} + (1 - \alpha) |d_{i-1} - n_{i-1}|; \]
    \[ \text{varfactor}_i = \beta/4 \ v_i; \]
    \[ D_i = n_i + \text{varfactor}_i; \]
    if ( \[ D_i < ARdelay_i + \beta v_i \] )
        \[ D_i = ARdelay_i + \beta v_i; \]
    end
else // Normal mode
    \[ ARdelay_i = \alpha ARdelay_{i-1} + (1 - \alpha) n_{i-1}; \]
    \[ v_i = \alpha v_{i-1} + (1 - \alpha) |d_{i-1} - n_{i-1}|; \]
    \[ \text{varfactor}_i = \beta v_i; \]
    \[ D_i = n_i + \text{varfactor}_i; \]
end

// if end-to-end delay < network delay
if ( \[ D_i < n_i \] )
    \( \text{packet}_i = \text{LOST}; \)
else
    \( \text{packet}_i = \text{IN\_TIME}; \)
end

if ( \[ n_i > d_i \] )
    \( \text{mode} = \text{NORMAL}; \)
end

if ( \[ n_i > d_i + 5v_i \] OR (packet\_i==\text{LOST}) )
    \( \text{mode} = \text{SPIKE}; \)
end
Chapter 4 – Control of the Playout Delay over a MANET

The last few years have been characterized by a rapidly growing market share of Voice over IP (VoIP) providers against traditional voice service operators, thanks to the low cost of the packet-based technologies and the reliability of the current (wired) IP networks. We believe that a similar success is expected to happen in mobile ad hoc networks (MANETs), which may offer a good platform for the fast deployment of VoIP mobile networks. However, efforts must be made to improve performance before MANETs can be used for this purpose. One of the main limitations is related to the highly variability of the network topology and channel behavior, which heavily influences the service quality due to route losses and significant delay variations.

It is easy to imagine a number of applications where this type of properties would bring benefits. One interesting research area is inter-vehicle communications. It is one area where the ad hoc networks could really change the way we communicate covering personal vehicles as well as professional mobile communication needs. Also, it is an area where no conventional (i.e. wired) solutions would do because of the high level of mobility. When considering demanding surroundings, say mines for example, then neither would the base station approach work but we must be able to accomplish routing via nodes that are part of the network i.e. we have to use ad hoc network. Such networks can be used to enable next generation of battlefield applications envisioned by the military including situation awareness systems for maneuvering war fighters, and remotely deployed unmanned micro-sensor networks. Ad Hoc networks can provide communication for civilian applications, such as disaster recovery and message exchanges among medical and security personnel involved in rescue missions.

In this chapter, it’s proposed a strategy where these impairments are jointly addressed. The source is responsible for jointly selecting the transmission paths and adjusting the playout delay, with an adaptive inter-talkspurt approach. These tasks are accomplished on the basis of historical data on network connectivity and transmission delays, and are driven by a quality-based approach. The collection of statistics of the network status relies on the QOLSR routing algorithm, whereas the voice quality is measured by means of the ITU-T E Model.

The chapter is organized as follows. First, we illustrate the Mobile ad Hoc Network. Then, we provide a description of the quality-based approach for adaptive dejittering in VoIP communications and a description of the main features of the chosen routing algorithms. The proposed strategy for joint routing and playout buffering is then described. Finally, we provide the simulation results.
4.1 An Overview of VoIP in Mobile Ad hoc Networks

MANET networks are defined as dynamic multi-hop wireless temporary networks, which are established by a group of mobile hosts on a shared wireless channel [71]. MANETs are often defined as follows: A "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links - the union of which forms an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. The strength of the connection can change rapidly in time or even disappear completely. Nodes can appear, disappear and re-appear as the time goes on and all the time the network connections should work between the nodes that are part of it. As one can easily imagine, the situation in ad hoc networks with respect to ensuring connectivity and robustness is much more demanding than in the wired case.

Ad hoc networks are networks not (necessarily) connected to any static (i.e. wired) infrastructure. An ad-hoc network is a LAN or other small network, especially one with wireless connections, in which some of the network devices are part of the network only for the duration of a communications session or, in the case of mobile or portable devices, while in some close proximity to the rest of the network.

The ad hoc network is a communication network without a pre-exist network infrastructure. In cellular networks, there is a network infrastructure represented by the base-stations, Radio network controllers and so on. In ad hoc networks every communication terminal (or radio terminal) communicates with its partner to perform peer to peer communication. If the required radio terminal is not a neighbor to the initiated call radio terminal (outside the coverage area of the radio terminal), then, the other intermediate radio terminals are used to perform the communication link. This is called multi-hop peer to peer communication. This collaboration between the RTs is very important in the ad hoc networks. In ad hoc networks all the communication network protocols should be distributed throughout the communication terminals (i.e. the communication terminals should be independent and highly cooperative).

MANET is a collection of independent mobile nodes that can communicate to each other via radio waves. The mobile nodes that are in radio range of each other can directly communicate, whereas others need the aid of intermediate nodes to route their packets. These networks are fully distributed, and can work at any place without the help of any infrastructure. The characteristics of these networks are summarized as follows:

- Communication via wireless means
- Nodes can perform the roles of both hosts and routers
- No centralized controller and infrastructure. Intrinsic mutual trust
- Dynamic network topology. Frequent routing updates
- Autonomous, no infrastructure needed
- Can be set up anywhere
- Energy constraints
- Limited security

Generally, the communication terminals have a mobility nature which makes the topology of the distributed networks time varying. The dynamical nature of the network topology increases the challenges of the design of ad hoc networks. Each radio terminal is usually powered by energy limited power source (as rechargeable batteries). The power consumption of each radio terminal could be divided generally into three parts, power consumption for data processing inside the RT, power consumption to transmit its own information to the destination, and finally the power consumption when the RT is used as a router, i.e. forwarding the information to another RT in the network. The energy consumption is a critical issue in the design of the ad hoc networks.

The mobile devices usually have limited storage and low computational capabilities. They heavily depend on other hosts and resources for data access and information processing. A reliable network topology must be assured through efficient and secure routing protocols for Ad Hoc networks.

Briefly summarizing, the main characteristics of this particular type of network are the absence of a centralized system, all nodes communicate to every other and they are in continuous movement. On the one hand, these characteristics make these networks very attractive, since almost no wired installations are required making very easy the provisioning of the communication services; on the other hand, these make the deployment of reliable and high-quality services very challenging. In this context, a great role is taken by the routing algorithm which is in charge of finding reliable routes among all the possible paths between the source and the destination, when network topology and channel performance change constantly.

To compare and analyze mobile ad hoc network routing protocols, appropriate classification methods are important. Classification methods help researchers and designers to understand distinct characteristics of a routing protocol and find its relationship with others. These characteristics mainly are related to the information which is exploited for routing, when this information is acquired, and the roles which nodes may take in the routing process.

One of the most popular methods to distinguish mobile ad hoc network routing protocols is based on how routing information is acquired and maintained by mobile nodes. Using this method, the proposed routing protocols can be classified in three categories [72, 73]: proactive, reactive and hybrid. The first builds routing tables exploiting a periodic exchange of topology information with other nodes in the network; the second determines routes only when needed; the last is a mix of the first two types of protocols.

A proactive routing protocol is also called "table driven" routing protocol. Using a proactive routing protocol, nodes in a mobile ad hoc network continuously evaluate routes to all
reachable nodes and attempt to maintain consistent, up-to-date routing information. Therefore, a source node can get a routing path immediately if it needs one. In proactive routing protocols, all nodes need to maintain a consistent view of the network topology. When a network topology change occurs, respective updates must be propagated throughout the network to notify the change. Most proactive routing protocols proposed for mobile ad hoc networks have inherited properties from algorithms used in wired networks. To adapt to the dynamic features of mobile ad hoc networks, necessary modifications have been made on traditional wired network routing protocols. Using proactive routing algorithms, mobile nodes proactively update network state and maintain a route regardless of whether data traffic exists or not, the overhead to maintain up-to-date network topology information is high.

Reactive routing protocols for mobile ad hoc networks are also called "on-demand" routing protocols. In a reactive routing protocol, routing paths are searched only when needed. A route discovery operation invokes a route-determination procedure. The discovery procedure terminates either when a route has been found or no route available after examination for all route permutations. In a mobile ad hoc network, active routes may be disconnected due to node mobility. Therefore, route maintenance is an important operation of reactive routing protocols. Compared to the proactive routing protocols for mobile ad hoc networks, less control overhead is a distinct advantage of the reactive routing protocols. Thus, reactive routing protocols have better scalability than proactive routing protocols in mobile ad hoc networks. However, when using reactive routing protocols, source nodes may suffer from long delays for route searching before they can forward data packets.

Hybrid routing protocols are proposed to combine the merits of both proactive and reactive routing protocols and overcome their shortcomings. Normally, hybrid routing protocols for mobile ad hoc networks exploit hierarchical network architectures. Proper proactive routing approach and reactive routing approach are exploited in different hierarchical levels, respectively.

The latest developments have increased the service retainability and accessibility in MANET networks, which represent now a viable solution for the deployment of networking services for data application at low costs and minimizing network installations. However, many efforts are still needed to make this technology workable for streaming applications where QoS requirements, in terms of packet losses, delay and jitter, are more stringent than those characterizing data applications. Only few works appeared in the last years on the analysis of quality issues in real-time traffic transmission over MANET networks. [74] evaluates the performance of different reactive routing protocols, such as the Ad Hoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR) and Temporally-Ordered Routing Algorithm (TORA), when varying the load of real-time traffic. These routing protocols introduce high transmission delays because they determine the routing table only if there is traffic in the network. In [75], the authors have analyzed the performance of VoIP systems in ad hoc networks with stationery nodes when using two routing protocols: AODV and Optimized Link
State Routing (OLSR). Also the authors in [76] have investigated the deployment of VoIP services with the AODV routing protocol and analyzed different performances metrics, such as jitter, one-way delay, frequency of service interruptions and their duration. These are the important performance metrics that affect the service quality experience of the users of mobile VoIP phones. Both papers have highlighted that the available routing algorithms still need to be improved to support telephony services over mobile ad hoc networks at a satisfactory quality.

Therefore, it is clear the intention of applying a quality-based model for VoIP, even in MANETs, such as that proposed in previous chapters.

Only few past researches entirely addressed the subject of this PhD study, yet, specific issues in VoIP over MANETs have been quite analyzed in some past works, which, briefly, I review in the following.

High packet loss rates caused by interference and fading in the wireless channels and excessive delay jitter caused by Carrier-Sense Multiple Access/Collision Avoidance (CSMA/CA) are the major obstacles against the deployment wireless VoIP services. Such impairments are further magnified in a Wireless Mesh Network (WMN) environment. Significant works in the area of VoIP over one-hop wireless networks exist, whereas the literature on VoIP in WMNs networks, and in particular in MANETs, is limited. The difference between WMN and MANET is only the mobility of nodes in MANET, but the way to improve the QoS is similar. In [77] and [78], the authors analyzed end-to-end performance enhancements when introducing a packet aggregating approach. In particular, [77] proposes a concatenation mechanism to reduce VoIP protocol overhead in a real multihop mesh network. These researchers have proposed three different algorithms, which differ on the basis of where the aggregator is located: end-to-end, where all packets towards a common destination are aggregated at the ingress node only; hop-by-hop, where aggregation and de-aggregation is done at every node, leading to better aggregation possibilities at the expenses of higher complexity and delay with respect to the first approach; accretion aggregation algorithm, which is a mix of the previous two techniques. The main benefit of these mechanisms is an increase in the number of calls that can be supported in the mesh network; however, the voice quality resulted to be affected by the aggregation. In [78], a different hop-by-hop aggregation mechanism is proposed. This technique significantly improves the performance of VoIP traffic meant as number of supported flows and at the same time reduces MAC layer busy time. But this mechanism has some drawbacks in terms of scalability; indeed, when the number of hops is low (up to five hops) the improvements of the aggregation algorithm with respect to the no-aggregation approach are negligible. Another approach to improve VoIP quality is to create differentiated services, which minimize starvation of best effort applications in the network [79]. [79] proposes the creation of Priority QoS Maps and indicates the difference such maps will have when compared to their Best Effort counterparts. By using Priority QoS Maps, an end user can immediately decide where to move to in order to obtain the acceptable QoS their VoIP application demands for. However, in this paper there isn’t any reference to a particular routing algorithm that may influence the
composition of the Priority QoS Maps. In [80], the authors made extensive experiments on a real WMN that supports VoIP, Internet and IPTV services. They have analyzed the performance in terms of delay and jitter at the increase of the service traffic load in the network. The result is that VoIP performs flawlessly over up to five hops, whereas some degradations are observed when there are any other types of traffic flows in parallel over the WMN, such as IPTV streams or voice conferencing in parallel over one hop. [81] proposes an Adaptive QoS Playout (AQP) algorithm to offer a high quality wireless VoIP service. AQP integrates the effect of dejitter buffer control, retransmission, and handoff delay based on a perceptual speech quality, the E Model. AQP first configures several buffer delays and evaluates the resulting packet loss rates at the beginning of each talkspurt. Then, it analyses the E Model quality index of these chosen delays to select the playout delay that optimize the expected quality for a VoIP session. The algorithm introduces packet retransmission which increases the delay; the resulting R factors values are around 63, which correspond to low voice quality. Additionally, when the end-to-end delay is large, the improvements of the AQP algorithm are not noticeable with respect to alternative adaptive algorithm and in this study the authors haven’t considered other traffic in the WMN.

4.2 Proposed Strategy

On the basis of these observations, we have investigated the introduction of a new strategy for VoIP packets routing, which takes into account the ITU-T E Model to evaluate the quality perceived by the end-user. The resulting algorithm relies on the Quality OLSR (QOLSR) extension of the OLSR protocol to allow the voice source to gather information about the network connectivity and transmission delays. Indeed, the QOLSR algorithm is used only to monitor the status of the network while the selection of the path is driven by the E Model for conversational quality evaluation. To improve the robustness, the proposed algorithm also exploits network diversity so that more than one path can be utilized in parallel; in this case, more copies of the same packets are sent through different independent paths. The selection of the most appropriate routes to the destination is performed according to an inter-talkspurt approach to minimize the impact of the service interruption on the voice quality. Additionally, routes selection is jointly performed with playout delay adjustments. These two tasks are performed together since we have observed that the setting of the playout delay heavily affects the benefit of the proposed routing strategy.

To implement the proposed strategy, a routing protocol that allows the source to know the network topology and the link transmission delays is needed. We haven’t defined any new routing protocol, but we have evaluated the plethora of existing ones and selected the one which can provide this information with only minor changes, that is the OLSR protocol and its
evolution QOLSR. Now, I give a brief description of these routing algorithms, but I subsequently will be treated in details.

Both these protocols, called proactive protocols, maintain routes to all destinations at all times through periodic advertisements. OLSR [82] is a table-driven proactive protocol which builds routing tables exploiting a periodic exchange of topology information with other nodes in the network. To create the routing table, this algorithm uses two particular control messages called Hello and Topology Control (TC) messages. Hello messages are used by each node to communicate its position only to nodes at one-hop and these messages are sent in broadcast. Each node then selects a sub-set of its neighbor nodes at 1-hop. These sub-set nodes, called Multi Point Relay nodes (MPR), are selected to announce the position periodically of its neighbor to other neighboring nodes through their TC messages. Thereby, each node announces to the rest of the network its adjacency information through its MPR. In route calculation, the MPRs are used to form the route from a given node to any destinations in the network. The protocol uses the MPRs to facilitate efficient flooding of control messages in the network. Indeed, this reduces the size of the control packets being flooded and, since only a subset of all nodes are selected as MPR, reduces the number of nodes flooding link-state information as well. As shown in Figure 4.1, in traditional routing any node communicates its reachability to the others by flooding control messages, with a great impact on the network traffic. In OLSR routing, the operation of the MPR nodes (black nodes in the figure) allows for a significant reduction of the routing information message exchange.

QOLSR is an enhancement of the OLSR routing protocol to support multiple-metric routing criteria. Each node calculates various quality metric (delay, bandwidth, loss probability, etc) in every neighbor link, through the transmission of Hello messages and its relative timestamps. Then, all metrics are included in the TC messages and sent to the other nodes to inform them. Moreover, metric informations are used to calculate sub-set nodes called QoS Multi Point Relay nodes (QMPR). Another difference between the two protocol versions is path selection: OLSR selects the shortest path considering the number of hops, whereas QOSLR selects the shortest paths on the basis of more metrics (delay, bandwidth, loss probability, and others).
4.2.1 Optimized Link State Protocol (OLSR)

Optimized Link State Protocol (OLSR) [83] is a proactive routing protocol, so the routes are always immediately available when needed. OLSR is an optimization version of a pure link state protocol. So the topological changes cause the flooding of the topological information to all available hosts in the network. To reduce the possible overhead in the network protocol uses Multipoint Relays (MPR). The idea of MPR is to reduce flooding of broadcasts by reducing the same broadcast in some regions in the network. Another reduce is to provide the shortest path. The reducing the time interval for the control messages transmission can bring more reactivity to the topological changes.

OLSR uses two kinds of the control messages: Hello and Topology Control (TC). Hello messages are used for finding the information about the link status and the host’s neighbors. With the Hello message the Multipoint Relay (MPR) Selector set is constructed which describes which neighbors has chosen this host to act as MPR and from this information the host can calculate its own set of the MPRs. The Hello messages are sent only one hop away but the TC messages are broadcasted throughout the entire network. TC messages are used for broadcasting information about own advertised neighbors which includes at least the MPR Selector list. The TC messages are broadcasted periodically and only the MPR hosts can forward the TC messages. There is also Multiple Interface Declaration (MID) messages which are used for informing other host that the announcing host can have multiple OLSR interface addresses. The MID message is broadcasted throughout the entire network only by MPRs. There is also a “Host and Network Association” (HNA) message which provides the external routing information by giving the possibility for routing to the external addresses. The HNA message provides information about the network- and the netmask addresses, so that OLSR host can consider that
the announcing host can act as a gateway to the announcing set of addresses. The HNA is considered as a generalized version of the TC message with only difference that the TC message can inform about route cancelling while HNA message information is removed only after expiration time. The link in the ad hoc network can be either unidirectional or bidirectional so the host must know this information about the neighbors. The Hello messages are broadcasted periodically for the neighbor sensing. The Hello messages are only broadcasted one hop away so that they are not forwarded further. When the first host receives the Hello message from the second host, it sets the second host status to asymmetric in the routing table. When the first host sends a Hello message and includes that, it has the link to the second host as asymmetric, the second host set first host status to symmetric in own routing table. Finally, when second host send again Hello message, where the status of the link for the first host is indicated as symmetric, then first host changes the status from asymmetric to symmetric. In the end both hosts knows that their neighbor is alive and the corresponding link is bidirectional. The Hello messages are used for getting the information about local links and neighbors. The Hello messages periodic broadcasting is used for link sensing, neighbor’s detection and MPR selection process. Hello message contains: information how often the host sends Hello messages, willingness of host to act as a Multipoint Relay, and information about its neighbor. Information about the neighbors contains: interface address, link type and neighbor type. The link type indicates that the link is symmetric, asymmetric or simply lost. The neighbor type is just symmetric, MPR or not a neighbor. The MPR type indicates that the link to the neighbor is symmetric and that this host has chosen it as Multipoint Relay.

The Multipoint Relays (MPR) is the key idea behind the OLSR protocol to reduce the information exchange overhead. Instead of pure flooding the OLSR uses MPR to reduce the number of the host which broadcasts the information throughout the network. The MPR is a host’s one hop neighbor which may forward its messages. The MPR set of host is kept small in order for the protocol to be efficient. In OLSR only the MPRs can forward the data throughout the network. Each host must have the information about the symmetric one hop and two hop neighbors in order to calculate the optimal MPR set.

Information about the neighbors is taken from the Hello messages. The two hop neighbors are found from the Hello message because each Hello message contains all the hosts’ neighbors. Selecting the minimum number of the one hop neighbors which covers all the two hop neighbors is the goal of the MPR selection algorithm. Also each host has the Multipoint Relay Selector set, which indicates which hosts has selected the current host to act as a MPR. When the host gets a new broadcast message, which is need to be spread throughout the network and the message’s sender interface address is in the MPR Selector set, then the host must forward the message. Due to the possible changes in the ad hoc network, the MPR Selectors sets are updated continuously using Hello messages.

The algorithm constructs the MPR set which includes minimum number of the one hop symmetric neighbors from which it is possible to reach all the symmetrical strict two hop
neighbors. The host must have the information about one and two hop symmetric neighbors in order to start the needed calculation for the MPR set. All the exchange of information is broadcasted using Hello messages. The neighbors which have status of willingness different than WILL_NEVER in the Hello message can be chosen to act as MPR. The neighbor must be symmetric in order to become an MPR. In order to exchange the topological information and build the topology information base the host that were selected as MPR need to sent the topology control (TC) message. The TC messages are broadcasted throughout the network and only MPR are allowed to forward TC messages. The TC messages are generated and broadcasted periodically in the network. The TC message is sent by a host in order to advertise own links in the network. The host must send at least the links of its MPR selector set. The TC message includes the own set of advertised links and the sequence number of each message. The sequence number is used to avoid loops of the messages and for indicating the freshness of the message, so if the host gets a message with the smaller sequence number it must discard the message without any updates. The host must increment the sequence number when the links are removed from the TC message and also it should increment the sequence number when the links are added to the message. The sequence numbers are wrapped around. When the hosts advertised links set becomes empty, it should still send empty TC messages for specified amount of time, in order to invalidate previous TC messages. This should stop sending the TC messages until it has again some information to send. The size of the TC message can be quite big, so the TC message can be sent in parts, but then the receiver must combine all parts during some specified amount of time. Host can increase its transmission rate to become more sensible to the possible link failures. When the change in the MPR Selector set is noticed, it indicates that the link failure has happened and the host must transmit the new TC message as soon as possible.

The host maintains the routing table, the routing table entries have following information: destination address, next address, number of hops to the destination and local interface address. Next address indicates the next hop host. The information is got from the topological set (from the TC messages) and from the local link information base (from the Hello messages). So if any changes occur in these sets, then the routing table is recalculated. Because this is proactive protocol then the routing table must have routes for all available hosts in the network. The information about broken links or partially known links is not stored in the routing table. The routing table is changed if the changes occur in the following cases: neighbor link appear or disappear, two hops neighbor is created or removed, topological link is appeared or lost or when the multiple interface association information changes. But the update of this information does not lead to the sending of the messages into the network. For finding the routes for the routing table entry the shortest path algorithm is used.

OLSR is also a flat routing protocol; it does not need central administrative system to handle its routing process. The proactive characteristic of the protocol provides that the protocol has all the routing information to all participated hosts in the network. However, as a
drawback OLSR protocol needs that each host periodic sends the updated topology information throughout the entire network, this increase the protocols bandwidth usage. But the flooding is minimized by the MPRs, which are only allowed to forward the topological messages. The reactiveness to the topological changes can be adjusted by changing the time interval for broadcasting the Hello messages. It increases the protocols suitability for ad hoc network with the rapid changes of the source and destinations pairs. Also the OLSR protocol does not require that the link is reliable for the control messages, since the messages are sent periodically and the delivery does not have to be sequential. Due to the OLSR routing protocol simplicity in using interfaces, it is easy to integrate the routing protocol in the existing operating systems, without changing the format of the header of the IP messages. The protocol only interacts with the host’s Routing Table. OLSR protocol is well suited for the application which does not allow the long delays in the transmission of the data packets. The best working environment for OLSR protocol is a dense network, where the most communication is concentrated between a large numbers of nodes. OLSR has also extensions to allow for hosts to have multiple OLSR interface addresses and provide the external routing information giving the possibility for routing to the external addresses. Based on this information there is possibility to have hosts in the ad hoc network which can act as gateways to another possible network.

### 4.2.2 Dijkstra Algorithm

Dijkstra Algorithm is one of the more common shortest path algorithms and can be applied to network routing. This Algorithm is an iterative process that works through a graph or a set of vertices & paths to calculate the shortest path from any one source node to every other node in the set.

The Dijkstra algorithm goes through these steps: the router builds a graph of the network and identifies source and destination nodes, as V1 and V2 for example. Then it builds a matrix, called the "adjacency matrix." In this matrix, a coordinate indicates weight. For example, [i, j] is the weight of a link between V_i and V_j. If there is no direct link between V_i and V_j, this weight is identified as infinity/∞.

- The router builds a status record set for every node on the network. The record contains three fields:
  - Predecessor field - The first field shows the previous node.
  - Length field - The second field shows the sum of the weights from the source to that node.
  - Label field - The last field shows the status of node. Each node can have one status mode: "permanent" or "tentative."
- The router initializes the parameters of the status record set (for all nodes) and sets their length to "infinity" and their label to "tentative."

- The router sets a T-node. For example, if V1 is to be the source T-node, the router changes V1's label to "permanent." When a label changes to "permanent," it never changes again. A T-node is an agent and nothing more.

- The router updates the status record set for all tentative nodes that are directly linked to the source T-node.

- The router looks at all of the tentative nodes and chooses the one whose weight to V1 is lowest. That node is then the destination T-node.

- If this node is not V2 (the intended destination), the router goes back to step 5.

- If this node is V2, the router extracts its previous node from the status record set and does this until it arrives at V1. This list of nodes shows the best route from V1 to V2.

These steps are shown below as a flow graph (Figure 4.2).

![Flow graph of Dijkstra Algorithm](image.png)
Now we want to find the best route between A and E (see below, Figure 4.3) through an example that is most enlightening. There are six possible routes between A and E (ABE, ACE, ABDE, ACDE, ABDCE, ACDBE), and it's obvious that ABDE is the best route because its weight is the lowest. But life is not always so easy, and there are some complicated cases in which we have to use algorithms to find the best route.

i. As in the image below (Figure 4.3a), the source node (A) has been chosen as T-node, and so its label is permanent (I show permanent nodes with filled circles and T-nodes with the --> symbol).

![Figure 4.3. a) Finding the best route. STEP 1](image)

Now we can have an adjacency matrix of the network (Table 4.1). If there is no direct link between two nodes, then assuming the weight is infinite.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>INFINITE</td>
<td>INFINITE</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>INFINITE</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>INFINITE</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>INFINITE</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>INFINITE</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

ii. In this step, the status record set of tentative nodes directly linked to T-node (B, C) has been changed. Also, since B has less weight, it has been chosen as T-node and its label has changed to permanent (see below)
iii. In this step, like in step 2, the status record set of tentative nodes that have a direct link to T-node (D, E), has been changed. Also, since D has less weight, it has been chosen as T-node and its label has changed to permanent (see below).

![Diagram](image1.png)

Figure 4.3. b) Finding the best route. STEP 2

iv. In this step, there are not tentative nodes, so it’s possible to identify just the next T-node. Since E has the least weight, it has been chosen as T-node.

![Diagram](image2.png)

Figure 4.3. c) Finding the best route. STEP 3

v. E is the destination, so we stop here.

There is the end. Now we have to identify the route. The previous node of E is D, and the previous node of D is B, and B’s previous node is A. So the best route is (ABDE). In this case, the total weight is 4 (1+2+1).

Alternative roots are:
\[
\begin{align*}
(ABE) & \rightarrow 1 + 4 = 5 \\
(ACE) & \rightarrow 5 + 3 = 8 \\
(ACDE) & \rightarrow 5 + 2 + 1 = 8
\end{align*}
\]

### 4.3 Playout Buffering

In the proposed algorithm, the voice source performs the most important operations. These can be summarized as follows: collecting information on the status of the MANET network and jointly computing the best routes and playout delays. These tasks are accomplished in parallel and continuously during the whole voice communication session. However, the adjustments of the routes and the playout delays are performed in an intra-talkspurt mode, that is, changes are introduced during silence periods only to reduce the impact of voice segments shrinking and/or expansion on the service quality.

As already anticipated in the previous, in this PhD study I refer to the MANET network technology and implementation of VoIP on top of it. The main characteristic of the MANET is the absence of a central node which manages the network communications; indeed, this network is set up by mobile peer wireless nodes, which may act as end-system in a communication and router in another. In this PhD study I refer to a particular MANET model called flat topology [84].

This topology is characterized by the complete absence of nodes groups with dissimilar roles in the communication; accordingly, any nodes have equivalent responsibility in managing the communications and all contribute to find the required radio link resources, at least. One of the advantages of this topology is the possibility to find many routes between two end-points on the basis of a predefined criterion, as proposed in this devised algorithm.

As already mentioned, VoIP applications have been developed over a set of protocols (RTP, UDP, and IP) that are not able to natively guarantee the application required quality of service. In fact, different factors deeply affect the end-user perceived quality. One of the most impairing factors is the variation of the packet transmission delay during the streaming, named jitter, which is caused by the temporal variability of the network conditions. Due to the high variability of the network topology and channel behavior, in MANET networks this impairment is of greater extent than in other types of networks (wired or infrastructured wireless networks). We propose to reduce the effects of this impairment by monitoring the network status and selecting the route that at the moment is characterized by low network delay and jitter. To improve the resulting quality, we also consider sending voice packets over more paths in parallel so as to increase the probability to receive the packet in time before its playout instant. Additionally, since the quality of a route selection depends on the playout delay, we propose to control both route selection and playout buffer size jointly. This increases the effectiveness of both actions. The selection of the optimal configuration is performed on the basis of the ITU-T
conversational quality model. It has the advantage to consider the most important network settings and performance metrics and to evaluate their impact on the voice quality taking into account the human perception. Whereas this quality-based approach has been already proposed in the past for playout buffering, it is new in driving the paths selection. The OLSR protocol provides the functionalities to allow each node to gather information about network connectivity. Since we also need delays statistics to evaluate the expected end-to-end quality of service, we assume that the network nodes run the QOLS protocol, which allows them to exchange information on communication quality metrics, such as delay, bandwidth, administrative link weights and so on. In this scenario we assume that every $T$ seconds, each node sends in broadcast different Hello messages to tell their adjacent nodes its position and the transmission delays. In this way, each node acquires information about the local connectivity. To widespread this information through the entire network, according to the QOLS algorithm TC messages are sent by the QMPR nodes. This message contains the list of nodes directly connected to the QMPR and the connection quality metrics, which in this study are just the observed link delays. The TC message contains the list of neighbors which have selected the sender node as a QMPR and their QoS constraints, in this case only delay of the nodes directly connected. The sequence number (ANSN) associated with this QMPR Selector set is also sent with the list. The information diffused in the network by these extension TC messages will help each node to calculate its routing table (Figure 4.4). Therefore, every node is able to build the whole graph of the network. Since QOLS is a proactive protocol, each change triggers an update of the topology information. By using the Hello messages to convey links delays, each node is able to build not only the oriented network graph but to associate delay vectors for each link. The setting of the $T$ parameter is very important to control the temporal resolution of the transmission delay statistics (the lower the value of $T$ the higher the resolution) and the traffic overhead (the higher the value of $T$ the lower the amount of additional traffic).

![Figure 4.4. TC-messages format in QOLS protocol](image)

Figure 4.5 shows the flow graph of the proposed algorithm. Network connectivity and transmission delay monitoring, as described previously, is performed during both talkspurt and silence periods and requires the cooperation of every node in the network. Differently, path
selection and playout delay setting is performed only during silence periods. There are many silence detection algorithms that can be used and that are implemented in most codecs to reduce the final source bit rate. This can be used to find the end and beginning of each talkspurt.

As already mentioned, the proposed strategy exploits path diversity by sending more copies of voice packets along different paths in parallel, if sending more copies is expected to bring a significant positive impact on the voice conversational quality. The existing paths between the source-destination couple are computed on the basis of a matrix representing the topology of the MANET. It is a time-variant matrix as each node may change its position and enter and leave the network dynamically. Let $M$ be this matrix at the end of a talkspurt. It is build on the basis of the links delays $d_j$ collected during the $K$ previous seconds of time over the entire network by means of the Hello and TC messages. This matrix is computed at the source by averaging the delays observed over all the links in the network. This matrix then provides also information on the connectivity of the nodes since if no delays statistics have been collected during this interval in a link, then the relevant nodes are assumed not to be directly connected.
Figure 4.5. Flow graph of our proposed algorithm

We then apply the Dijkstra’s algorithm to matrix $M$, which is a greedy algorithm that solves the single-source shortest path problem for a directed graph with non-negative edge weights. The Dijkstra’s algorithm is initially used to compute the shortest path; then, the relevant links are removed from matrix $M$ and the algorithm is again applied. As a final result we obtain a vector $P$ of paths. These steps are performed iteratively, so as to obtain $N$
independent paths, which are then stored in matrix $P$. The example in Figure 4.6 illustrates the coding of the $P$ and $M$ matrices.

Figure 4.6. (a) Simple network made of 8 nodes; (b) relevant matrix $M$; (c) matrix $P$ of paths between the source-destination pair 6-5.
The reference network is formed by 8 nodes (Figure 4.6(a)) and the relative matrix $M$ is represented in Figure 4.6(b), where $d_{ij}$ represents the delay between nodes $i$ and $j$. If two nodes are not visible each other, the relative delay is $\infty$. If 5 and 6 were the two communicating nodes, the proposed algorithm to compute the source-destination paths brings to 3 paths which are coded in matrix $P$, as represented in Figure 4.6(c).

At this point, we need to elaborate again the delay values for the links that take part in paths $P$. This time we consider all the delay values collected from the beginning of the voice session with the aim to estimate the probability density function (pdf) of the link transmission delay. Let $j$ index the links in the paths in $P$ and $f_j(d)$ be the relevant pdf. There are several algorithms that can be used to build and update $f_j(d)$ from historical data; some effective solutions are presented in [85].

In this PhD study, I make use of the flush and refresh approach; however, more complex techniques that take into account the age of the collected statistics can also be used to improve the pdf prediction accuracy. $f_j(d)$ are then used to computed $f_i^P(d)$, which is the pdf of the delays trough path $i$ in $P$ ($i=1,...,N$). Each $f_i^P(d)$ is obtained by computing the convolution of all the $f_j(d)$ relevant to the links in path $i$. The result of the convolution is a good estimation of the pdf of the path delays as far as the delays encountered by a packet from a link to another are independent. Indeed, while this may be true in many scenarios, there are some circumstances that can affect this assumption, e.g., frequent collisions cause high transmission delays in adjacent wireless links. It follows the estimation of the loss rate function $L_i(d)$ for every path $i$ varying the playout delay in a range such that the loss rate varies from 0 to 0.05. We also consider all the possible combinations of paths that can be used to send copies of the same packet in parallel. We start from the combinations of two paths, then three, and so on till the combination that includes all the $N$ paths. We index all the possible combinations with $h$ ($h=1,...,H$), where $H$ is the maximum number of combinations made of 1, 2, 3,.., $N$ paths together. Note that $H$ includes also the $N$ single-path solutions. We then compute $L_h^c(d)$, which is the expected packet loss rate for any solution $h$. When more copies of the same packet are sent through different paths, the loss rate is obtained considering the probability that none of the copies sent in parallel arrives before the playout delay. The loss rate functions are then used in the E Model to evaluate the performance expected when using each of the solutions $h=1,...,H$. For any one of these possible solutions, we compute the playout delay that brings to the maximum R factor value: $P_{D,h}^*$. While the use of more paths together allows for higher R factor scores, we take into account the cost associated to the number of paths used with function $C(#\_\_\_\_\_\_\text{copies})$. Let $c_h$ be the number of paths in combination $h$, the cost is computed for every solution as follows:
and we select the combination $h^*$ which maximizes previous formula of $F_h$. The path(s) in solution $h^*$ are then used during next talkspurt and the playout delay is set to $P_{D,h^*}$.

### 4.4 Measurements and Results

We have performed extensive experiments using the Network Simulator 2 (NS2 v.2.29) environment [86] with additional components simulating the OLSR protocol and introducing the required changes to reproduce the transmission of the QOLSR messages. NS2 is a discrete event simulator targeted at networking research. NS2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. To build NS2, it’s need a computer and a C++ compiler.

We considered the MANET IEEE 802.11b (11 Mbps) standard and simulated a wireless network with 10 nodes moving in an open area of 400m x 400m. Each node moves at the average speed of 5km/h in a flat country without any type of obstacle; each node generates background traffic directed to other 3, 4 or 5 randomly selected stations at a rate of 500 kbps with IP packets of average length of 500Bytes and characterized by an interarrival time exponentially distributed with an average of 8ms. Hello and TC packets are generated every 1sec and 5sec, respectively, to create the routing table. All simulation runs last 200sec and the $M$ matrix is updated considering the 5 previous seconds ($k=5$sec). Table 4.2 summarizes the main settings.

**Table 4.2. Simulation configuration parameters**

<table>
<thead>
<tr>
<th># nodes</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Area</td>
<td>400m x 400m</td>
</tr>
<tr>
<td>MAC type</td>
<td>802.11b</td>
</tr>
<tr>
<td>Simulation run length</td>
<td>200sec</td>
</tr>
<tr>
<td>Hello packets interval (T)</td>
<td>1 sec</td>
</tr>
<tr>
<td>TC packets interval</td>
<td>5 sec</td>
</tr>
<tr>
<td>Background traffic rate</td>
<td>500kbp</td>
</tr>
<tr>
<td>Background packets dimension</td>
<td>500Bytes</td>
</tr>
</tbody>
</table>
Once the algorithm has generated the average-delay-weighted graph and matrix $M$, it selects $N=5$ shortest paths through the use of the Dijkstra’s algorithm to build matrix $P$. $N$ has been set to 5 to select a significant number of paths but we then consider at most 3 paths in parallel to transmit voice packets. Indeed, three paths in parallel are considered to be the upper bound in bandwidth consumption for voice communications, considering that the three paths that would be selected are the most reliable in terms of losses and jitter. This prevent from making the search of the optimal solution too complex without any foreseen advantage. With this parameter, the search space is given by: 5 single-path solutions, 10 double-path solutions, 10 triple-path solutions, with a total of $H=25$ solutions to be analyzed.

In the following, we present a comparison of the results for three algorithms:

- OLSR algorithm combined with the quality-based play out control algorithm: we refer to this solution with OLSR Path. It means that the transmission path is selected according to the OLSR algorithm on the basis of the number of hops for any possible path and the playout buffering is controlled by maximizing the expected quality using the quality-based dejittering approach.
- Proposed algorithm without path diversity: we name this solution Single Path.
- Proposed algorithm with path diversity: we refer to the relevant results indicating the value of the cost function, $C(\text{_copies})$, when using two or three paths in parallel. Note that $C(1)$ has always been set to zero.

The implementation of the two versions of the proposed algorithm allows me to evaluate the benefits coming from using multiple paths for the same voice session. Figures 4.7 and 4.8 show the results of the three techniques at varying settings of the cost function. The following combinations are reported in these figures: $C(2)=0$, $C(3)=0$; $C(2)=3$, $C(3)=6$; $C(2)=13$, $C(3)=16$. With the first setting, increasing the number of paths selected doesn’t reduce $F_h$, meaning that the algorithm will select always the most appropriate combination of three paths.
As to the second setting, the algorithm selects double-path solutions in case the resulting R factor is higher than three points with respect to that of single-path solutions and selects triple-path solutions in case the R factor improvement is higher than six points. Similar considerations can be done for the third setting of the cost function.
Figure 4.7 shows clear results on the performance of the three algorithms. With the OLSR Path, the selected path remains the same for the entire session; this is the shortest one, at least at the beginning of the session, and the adaptation of the dejitter buffer to maximize the end-to-end quality allows to keep the quality almost constant during the entire session. As to the Single Path strategy, the varying selection of the routing path doesn’t always bring to the expected results with performance lower than the OLSR Path, as it happens frequently during the second half of the session. The reason for this degradation is probably due to the fact that the link delays are collected at a coarse resolution (1sec), which doesn’t allow for obtaining a reliable estimation of the path performance in terms of delays. The use of more paths in parallel allows for compensating this problem. As it can be observed from the figure, this algorithm allows to significantly increasing the voice quality of more than 20 points, as it is shown in Table 4.3. As expected, the lower the costs associated to the number of paths, the higher the performance. Figure 4.8 shows the playout delay. It can be noted as the OLSR path always select a high playout delay with a very low variance. This is because the selected path is not characterized by significant changes during the sessions. Differently, the Single Path introduces significant changes in the playout delay to follow the changing characteristics of the selected path. The use of more paths in parallel allows to use low playout delays with low variance thanks to the compensation of the transmission delay encountered by packets following dissimilar paths in parallel.

Table 4.3. Playout delay and R-Factor: average and variance

<table>
<thead>
<tr>
<th></th>
<th>average $D$ (sec)</th>
<th>$\sigma_D^2$</th>
<th>average $R$ (sec)</th>
<th>$\sigma_R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi Path C=0</td>
<td>0.212</td>
<td>0.00016</td>
<td>72.52</td>
<td>5.607</td>
</tr>
<tr>
<td>Multi Path C(2)=3; C(3)=6</td>
<td>0.218</td>
<td>0.00019</td>
<td>71.15</td>
<td>5.861</td>
</tr>
<tr>
<td>Multi Path C(2)=13; C(3)=16</td>
<td>0.247</td>
<td>0.00154</td>
<td>65.24</td>
<td>38.54</td>
</tr>
<tr>
<td>Single Path</td>
<td>0.273</td>
<td>0.00391</td>
<td>53.32</td>
<td>101.6</td>
</tr>
<tr>
<td>OLSR Path</td>
<td>0.397</td>
<td>-0</td>
<td>51.67</td>
<td>0.004</td>
</tr>
</tbody>
</table>
In this PhD thesis, we have presented a research work to improve the Quality of Service (QoS) in real-time streaming system mainly on VoIP on NGNs, and the definition of test setting for the measurement of quality parameters, considering that one of the main limitations is related to the highly variability of the network topology and channel behavior, which heavily influences the service quality due to route losses and significant delay variations. We have focused on the problems of jitter, packet loss and delay, which are the three main network-related Key Performance Indicators for QoS of VoIP. Mainly the jitter is an innate and damaging factor in IP Telephony. It is caused by the high variability of the network component of the delay, caused by some underlying time-variant network factors, such as network congestion, improper queuing, or configuration errors. The jitter causes irregular inter-packet delays at the destination side and this is unacceptable for a correct decoding and playout. To solve this problem, a playout control mechanism is used at the receiver side to compensate the jitter at the expense of an additional delay (introduced by the buffer). The intent is to obtain a new sequence of transport packets, uniformly spaced and decoded. The logic is quite simple: stop the received packets in the buffer for a certain interval of time in order to reconstruct the exact order (identical to the sender side) of the transport packets. Usually a playout mechanism sets the end-to-end delay, which includes also the playout delay (i.e., the dimension of the buffer). This setting is performed several times during an IP Telephony session and several playout control strategies exist that perform this task with different approaches.

My research studies are focus, exactly, on the problem of playout buffer sizing for IP Telephony services. The setting of the size of the buffer is a key operation for it directly affects the packet loss rate and the conversational interactivity, which quite influence the quality perceived by the end-user. Buffer adjustments may be introduced to avoid packet losses at all, to keep the packet loss under a desired threshold, or to have the optimal balance between loss and delay on the basis of a reference objective quality model.

On the basis of this consideration, the playout buffering algorithm estimates the optimal buffer configuration by weighting the contribution of delay and loss to the conversational quality. The use of such a perceptually motivated optimality criterion allows the receiver to automatically balance packet delay versus packet loss and it is driven by a quality-based approach. The buffer size is adaptively adjusted so that the expected quality during the next conversation period is maximized. We have made use of the ITU-T E Model for quality evaluation. It is a computational framework for the estimation of the conversational quality by
means of a synthetic index (the R factor), which encloses the contributions of many features as delay and packet loss, presented as impairment factors.

After analyzing the various existing standards and selected those considered most performance, we have applied this technique that maximize the perceived quality, to two particular networks: GEO Satellite Network and Mobile Ad hoc Network.

In the first network, a playout control algorithm for communications over satellite networks, named Ad-Hoc Sat, has been presented for VoIP services. The major contributions are the study of the GEO frame model and the proposal of a strategy that improves the end-user perceived quality by taking into account the delay statistics encountered by the RTP packet along the satellite channels. The performed tests show encouraging results when comparing the proposed technique with two alternative algorithms that include spike detection strategies.

In the second network, we have addressed the problem of packet routing and dejitter buffering in VoIP sessions over MANET networks. We propose a strategy where these impairments are jointly addressed. The source is responsible for jointly selecting the transmission paths and adjusting the playout delay, with an adaptive inter-talkspurt approach. These tasks are accomplished on the basis of historical data on network connectivity and transmission delays, and are driven by the quality-based approach. The collection of statistics of the network status relies on the QOLSR routing algorithm, whereas the voice quality is measured by means of the ITU-T E Model. The major idea is to consider the transmission of multiple copies of the same packet through different paths at the same time depending on the network status. The use of more paths in parallel is weighted with appropriate costs so that this option is used only when a single path would bring to poor voice quality. The experimental results showed a significant improvement of more than 20 points in terms of R factor.

This quality-based technique offers excellent performance and can be thought to apply it to other networks, due to its versatility. In fact this quality-based strategy is independent of the networks but only needs to be adapted to the defined network.

For this reason, in the future we intend to continue our investigation by focusing in another important networking system: the Unified Communication.

In the next section we want to explain a brief description of our investigated Unified Communication.

5.1 Unified Communication

In our increasingly mobile world, communication must be effective, global, and available through multiple technologies seamlessly. Unified Communications (UC) logically blends and combines previously separate services and features, making communication possible by any means, with anyone, using any of your devices.
UC integrates real-time and non real-time communications with business processes and requirements based on presence capabilities, presenting a consistent unified user interface and user experience across multiple devices and media types. UC supports the enterprise to manage various types of communications across multiple devices and applications, and across geographies, with personalized rules and policies, while integrating with back-office applications, systems and business processes. UC enables people to connect, communicate and collaborate seamlessly to improve business agility and results. These results include better user and group productivity, dynamic collaboration and simplified business processes, with the goal of increasing revenues, decreasing costs and improving customer service.

UC is not a single product but rather a solution made up of a variety of communication tools and components. Unified messaging, call control or IP communications is one element of a UC solution, but it is not UC in and of itself. UC is a comprehensive solution that ties several components together with user experience. UC components include:

- Call control and multimodal communications
- Presence
- Instant messaging
- Unified messaging
- Speech access and personal assistant
- Conferencing -- audio, Web and video
- Collaboration tools
- Mobility
- Business process integration (BPI)
- Software to enable business process integration

These UC tools are tied into business processes and applications, making the integrated solution exponentially more useful to businesses and workers. One key part of UC is called presence. Presence enables you (or software applications) to determine whether someone is available to communicate—either by telephone, instant message, Web sharing or even mobile phone. This makes communications much more efficient and greatly reduces "telephone tag." A typical UC session might start with an instant message between two parties that escalates to a phone call or Web conference through a click of a button on the PC screen. That click connects the parties via audio, and another turns the call into video, if desired. If other people need to be added to the conversation, a look at the presence status of people on your buddy list lets you simply click-to-conference to bring them into the call.

Unified communications is not a single product, it is a solution made up of various components, including a Call control/IP PBXs. While several vendors consider the switch or IP PBX to be the main element of a UC solution, and some consider UC to be merely an extension of Internet telephony, we view the IP PBX as a UC enabler, no more, no less. The PBX/IP PBX provides the plumbing needed for a UC solution. The IP PBX market is in a state of change, as vendors move toward software approaches and away from hardware-centric products. Service-
oriented architecture (SOA) and Web services are playing an important role in UC solutions, and most switch vendors have announced plans to provide call control capabilities via software rather than hardware.

However, the UC have still several problems, including: the high cost, the lack of integration with existing network architecture, and the voice quality perceived, that is not yet very good. In this context, my recent research has posted.

My focus is to the integration between different proprietary systems in a single quality-oriented platform. In this context my research has led to the integration of two systems that are gaining much success in the ICT community, the Microsoft Office Communications Server 2007 [87] and Digium’s Asterisk. The first is the system created by Microsoft Unified Communications while the second is an IP-PABX (Private Automatic Branch eXchange over IP network) open source. Asterisk has already been mentioned in previous sections, but now it will give a brief explanation of Microsoft Office Communications Server 2007 (OCS 2007).

OCS 2007 is one of the critical components of Microsoft’s Unified Communications strategy. With OCS 2007, companies can deploy an enterprise class instant messaging (IM), presence, web conferencing, and voice over IP (VoIP) solution to support the communications requirements for the organization. One of the aims of our investigation revolves around this last feature, improved quality VoIP. Although Office Communications Server 2007 is not a IP PBX, a user is able to use an Office Communications Server endpoint to communicate with the telephony world in a powerful and intuitive way by integrating with traditional telephony equipment (Figure 5.1). Microsoft’s unified communications leverages standards and published interfaces to interoperate and integrate with existing telephony and applications infrastructure investments, offering a flexible integration of telephony with other business communications tools.

![Diagram](image_url)

Figure 5.1. Integration between OCS 2007 and an IP PBX

The element that fulfills the telephony intermediation function with IP PBX has become the Office Communications Server role of Mediation Server.
With integration of these two programs, we are obtained one communication system with high performances, scalable, flexible, and with a strong reduction in costs. The integration between these systems necessarily lead to an analysis of performance of QoS and the use of algorithms for selecting the best quality, typically through the design of new algorithms dejittering. To allow an objective assessment of the reliability and validity of the project is necessary to define accurately all the tests through the use of simulators. This activity is currently under development.
Acknowledgements

In this thesis, the work that I conducted during my PhD activity at the CNIT Multimedia Communications Laboratory at the University of Cagliari was presented. This PhD research activity has been carried on within the IKNOS project funded by the Italian Ministry of Education and Research and developed by the University of Cagliari, CNIT and Tiscali.

Special thanks goes to Dr. Ing. Luigi Atzori, especially for the constant support in all my activities and valuable working advice.
References


[17] ETSI EN 300 969 - Half rate speech transcoding (GSM 06.20 version 8.0.1 Release 1999)

[18] ETSI EN 300 726 V.8.0.1 (2000-11) - (GSM 06.60 version 8.0.1 Release 1999)


[60] CNIT, Skyplex: CNIT satellite network, 2005


[64] http://www.asterisk.org/


