Dynamic Adaptation of Quality of Service for VoIP Communications

Nelson Costa
Instituto Superior Técnico
Lisbon, Portugal
eng.ncosta@gmail.com

Mário Serafim Nunes
IST, INESC-ID
Lisbon, Portugal
mario.nunes@inesc-id.pt

***Abstract* — The present work proposes an adaptive solution to provide quality of service in Voice over IP communications. This solution is based on three components that interact in order to achieve higher quality in voice communication. The first two consist in changing the codec and the transport protocol in real-time during a conversation; the third consists in using a Forward Error Correction mechanism to recover from loss packets. To demonstrate the voice quality obtained by this solution, a VoIP client application was developed, compatible with other VoIP clients, to implement the proposed quality of service algorithm and control the voice quality during a conversation. The results of the experimental measurements and simulations performed demonstrate that this solution is viable and significantly increases the voice quality of VoIP communication.**

Keywords: VoIP, QoS, Codec, SIP, RTP, RTCP, FEC, MOS, E-Model

I. INTRODUCTION

In recent years Voice over IP (VoIP) has proved to be a serious competitor to Public Switched Telephone Network (PSTN), in terms of cost, efficiency, versatility and reliability. VoIP started to be used by simple applications mainly to establish voice calls between computers and then evolved to establish calls through gateways to public telephone networks. Along this evolution, VoIP made significant steps to gain the acceptation of the business community, as it provides a telephone system over IP networks just for a fraction of the cost of the traditional telephone system.

Although VoIP is a convergence solution of data and voice networks and have a large group of advantages, there are still some aspects that need to be improved, such as quality of service (QoS). Classic telephone networks that use the circuit switching service have dedicated channels for each voice session and were planned to provide a deterministic quality of voice communication. IP networks use a Best Effort service that does not provide any QoS guarantee mechanism and where all traffic is assigned the same level of priority. According to the Best Effort service, the data sent by each user is processed and forwarded through the network according with the available bandwidth. When congestion is present in the network, the packets are discarded without any distinction, not guarantying that the service is carried through successfully. Although there are applications that are not affected by the delay, such as electronic mail, file transfer and web applications, there are some applications, as voice and video, that have stringent delay requirements, which normally are not guaranteed by the public Internet. It is imperative that interactive traffic receives a higher priority in relation to traditional traffic. There are several solutions aiming to provide some level of quality of service, such as, over-provisioning, IntServ [2], DiffServ [3] and MPLS [4], that introduce an additional complexity and cost to the network and are not yet widely deployed.

The main challenge in VoIP is to provide a high level of quality of service and being resilient to communication gaps. This challenge is still more difficult in wireless scenarios, where there is usually a high level of packet losses.

This paper presents an adaptive QoS solution to resolve the problem of the sporadic voice gaps that result from packet losses and variable delay during a conversation.

A VoIP client application was developed to test the quality of service of the proposed algorithm and control the quality of the voice provided during a conversation. This application is adaptable to the characteristics of the VoIP connection and compatible with other standard VoIP applications.

II. RELATED WORK

There are many papers that address QoS issues on VoIP, with different perspectives and applications. We selected for our analysis in particular the papers that address adaptive VoIP techniques.

In [5] is presented a method for evaluating various playout algorithms that extends the E-model by estimating user satisfaction from time varying impairments. The paper evaluates several playout algorithms and shows a correspondence between their results and those obtained via statistical loss and delay metrics.

In [6] is proposed an adaptive QoS playout algorithm based on the E-model, with dynamic retransmission in order to reduce the packet losses and in this way to increase the quality of the voice. The simulation results show improvements in the R-factor, specially for networks with low end-to-end delays, as LAN or WLANs. In the paper the authors do not considered to adapt the codec bitrate or the transport protocol.

In [7] an adaptive VoIP algorithm is proposed to improve the voice quality, based on several mechanisms, namely in the switch of the codec, FEC configuration and playout buffer size. The paper presents a theoretical analysis of the quality assessment methods and pseudo-code of the proposed algorithms, but the authors do not present an evaluation of the algorithms described, and consequently not allowing to assess the proposals presented.

In [8] an adaptive codec switching VoIP application over heterogeneous networks is presented, where the codec switching is based on the percentage of packet losses. An evaluation of the algorithm is presented, based on subjective MOS measurements, which shows that in most of the cases an increase in the voice quality is perceived when the algorithm is applied. However the proposed algorithm do not take in consideration the delay and jitter to switch the codec and consequently do not comply entirely with the E-model.

In [1] the authors of this paper present the basic concept of an adaptive VoIP algorithm fully compliant with the E-model that will be described and evaluated in detail in the following sections.

III. VoIP Techniques and Protocols Overview

Before describing the proposed adaptive quality of service solution for VoIP communications, it is necessary to introduce some definitions, technologies and protocols that support VoIP service and that will be mentioned along the paper.

A. *Voice Coding Techniques*

The sound produced by human's voice is defined by the vocal system, which comprises vocal cords, lips and nose. The air passing through the vocal system causes vibrations of the vocal cords that are modeled by the positions of the tong and lips. These vibrations are relatively slow, contributing to the predictability of the sound during the conversation. This predictability can be useful when a voice coding device named codec is applied, to minimize the quantity of data transmitted and necessary bandwidth. Codecs contain real-time compression algorithms that transform the analog signal of human voice into a bit sequence (digital signal) to be transmitted through the network, and the opposite functions at the receiver. The compression method removes the redundant and irrelevant components of the information, transmitting only what is essential to the communication. There are three types of codecs: Waveform codecs, Vocoders and Hybrid codecs. Waveform codecs try to play the original voice signal, sample by sample, based on its statistic, spectral and temporal characteristics. Vocoders codify only the important perceptual information of the voice. Hybrid codecs are a mix of the Waveform codecs and Vocoders.

One of the problems introduced when applying a voice compression system is to evaluate the voice quality. The Codec efficiency is related with the quality of the signal recovered by the destination, the used bandwidth and the algorithm complexity.

Table 1 shows the main characteristics of several codecs used in VoIP applications.

TABLE I. Audio Codecs Comparison

Codec	Algorithm	Sample Rate [KHz]	Transmission Rate [Kbit/s]	Delay [ms]
DVI4	ADPCM	8/16	32	?
G.711	PCM	8	64	0.125
G.723.1	ACELP/MP-MLQ	8	5.3/6.3	37,5
G.729	CS-ACELP	8	8	15
GSM FR	RPE-LTP	8	13	20
iLBC	LPC	8	13.33/15.2	25/40
Speex	CELP	8/16/32	[2.15-24.6][4-44.2]	30/34

B. *Quality of Service Parameters and Metrics*

The quality of service provided to applications is defined in terms of a Service Level Agreement (SLA), a specification that defines the QoS parameters thresholds. The more frequent QoS parameters used are: bandwidth, delay, jitter and packet loss.

- Bandwidth is the most basic QoS parameter that defines the minimum bitrate that the network needs to guarantee in order to be able to support the voice service.
- Delay represents the end-to-end delay and is calculated by the sum of all delays introduced by the equipments and the network.
- Jitter corresponds to the variation in time of data delivery, being caused by the variable delay in the network.
- Packet Loss Ratio (PER) represents the percentage of packet losses in the network.

In order to evaluate and measure permanently the quality of voice during a conversation as perceived by the user, we adopted the objective metric E-Model. The E-Model, defined in the ITU-T G.107 [9], analyses the QoS parameters and quantify the voice quality by calculating a scalar factor between 0 (worst case) and 100 (excellent), which is called the R factor. This factor represents the sum of all degradation factors in the communication, as shown in Fig. 1.

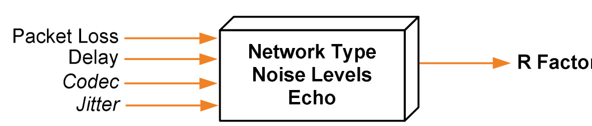

Figure 1. R factor calculation process

The E-Model R factor is obtained using the following expression:

$$R = (R_0 - I_s) - I_d - I_e + A \quad (1)$$

- Signal-to-Noise Ratio [R0], results of several types of noise, like transmission circuit noise, environment noise both in the transmitter and receiver, and a noise ceiling corresponding to the sensitivity of the human hearing system. ITU-T G.107 specification, defined the value 94.77 for this factor.

- Simultaneous Impairment [Is], represents all the impairments that occur simultaneous with the voice signal, like the quantization distortion caused by digitizing the voice signal. ITU-T G.107 specification defined the value 1.41 for this factor.

- Delay Impairments [Id], represents the losses associated to the end-to-end delay. I_d is obtained through the following expression:

$$I_d = I_{dte} + I_{dle} - I_{dd} \tag{2}$$

I_{dte} represents the transmission losses due to echo, I_{dle} the receiver losses due to echo and I_{dd} the voice absolute delay. ITU-T G.107 specification defines the expressions to calculate each I_d component.

According to Clark [10], it is possible to obtain the I_d value through an interpolated expression of those expressions. Although, according to Lustosa [11], that expression is incorrect, because it penalize the I_d factor for $T_a <= 175 \ ms$. This may be due to a typographical error and the correct expression is presented in the following interpolated expression:

$$\begin{cases} T_a \le 175ms : I_d = 0{,}023 T_a \\ T_a > 175ms : I_d = 0{,}111 T_a - 15{,}444 \end{cases} \tag{3}$$

T_a represents the system absolute delay and is obtained through the following expression:

$$T_a = T_{net} + T_{codec} + T_{buffer} \tag{4}$$

T_{net} represents the jitter, T_{codec} the codec delay and T_{buffer} the receiver buffer used for jitter compensation.

Expression 3 is represented graphically in Fig. 2.

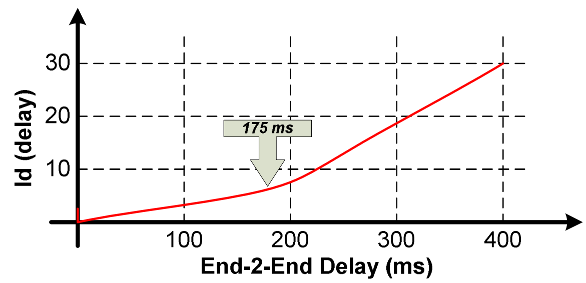

Figure 2. Relationship graphic between Id and delay

- Equipment Impairment [Ie], represents the equipment impairments caused by the codec and packet losses. ITU-T G.113 [12] defines a set of provisory values for this factor. Those values were converted to the graphics represented in Fig. 3 and Fig. 4.

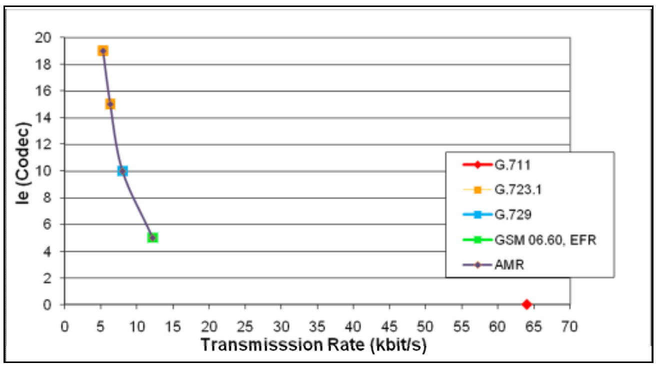

Figure 3. Ie provisory values as function of the codec

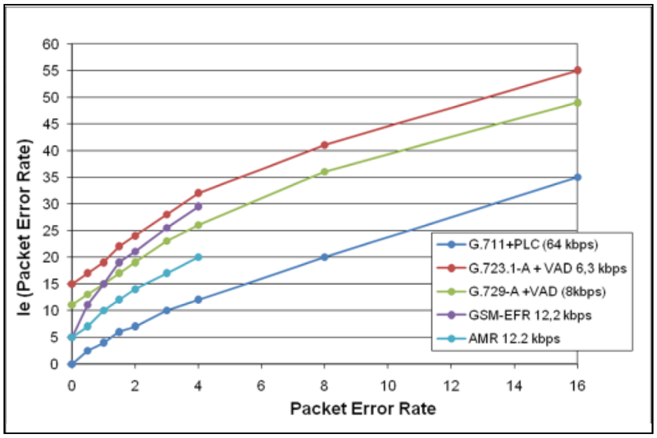

Figure 4. Ie as function of packet losses for different codecs

- Advantage Factor [A], allows for compensation of impairment factors where there are considered advantages of access to the user, e.g. mobile terminals. ITU-T G.113 specification defines A = 0 for fixed telephone networks.

According these provisional and standards values, the equation 1 can be reduced to the following expression:

$$R = 93{,}36 - I_d - I_e(codec, PER) \tag{5}$$

To categorize the results of the R factor on the perceived call quality, it was defined five categories of values, as showed in table 2.

TABLE II. R FACTOR CATEGORY VALUES

R Factor	User Satisfaction Level
90 ≤ R < 100	Very satisfied
80 ≤ R < 90	Satisfied
70 ≤ R < 80	Some users satisfied
60 ≤ R < 70	Many users dissatisfied
0 ≤ R < 60	Nearly all users dissatisfied

According to the limits defined in table 2, it is not recommend having an R ≤ 60, it is desirable to obtain R ≥ 70.

C. *Forward Error Correction*

The communication channel that transmits the signal containing the information could corrupt it by adding distortions and noise along the transmission. The exponential increase in the use and consumption of multimedia information through the wireless network, leads to maximize the bandwidth of the channel in order to maintain an acceptable quality of service to wireless network [13]. Since the wireless network communication channel, typically, has high error rates, caused by attenuation, dispersion and interference from other active sources, the main challenge is to provide a satisfactory service for multimedia communication on a channel with high error probability. In such cases it is imperative to use specific techniques to reduce these harmful effects and ensure a satisfactory level of voice quality in VoIP communications. In cases where selective retransmission is not possible due to stringent delay requirements, it is necessary to use redundancy mechanisms in order to recover from lost packets. This type of mechanisms is based on the transmission, in a controlled way, of redundant information that can be used in the receiver to correct possible errors that occur in the network.

Complementing the FEC technique, the receiver can also send a NAK (Negative acknowledge) when the data can not be recovered, reducing the number of retransmissions and energy consumption resulted from the retransmission of packets, since the energy consumption is higher in the transmission that in the reception.

FEC efficiency directly depends on the amount of redundant information added. Consequently it is necessary to estimate the error rate on the network to determine the amount of redundant information to be able to recover corrupt or lost packets.

D. *SIP, RTP and RTCP*

SIP, defined by IETF in RFC 3261 [14], is a text-based protocol, similar to the HTTP and SMTP, being designed for initiating, maintaining and terminating interactive communication sessions between users. Such sessions include voice, video, chat, interactive games, and virtual reality.

RTP, defined by IETF in RFC 3550 [15], defines a standard packet format to transport audio and video through Internet. The main reason for the definition of this protocol was to introduce temporal and sequential information that allow to identify the instants of emission and reception of data packets, as well as packet losses, information that is considered relevant when transferring real time data.

UDP is usually the transport protocol used to encapsulate the RTP packet because the applications that use RTP are more sensitive to delay than packet loss. However, UDP is a connectionless protocol that does not guarantee the delivery of packets and consequently the quality of service. However, there are situations where the use of TCP protocol could be better than UDP, namely in high packet loss environments, where the delay introduced by TCP could have lower impact in the voice quality than the delay caused the TCP retransmission mechanisms.

RTCP, defined by IETF in RFC 3550 [15], works with RTP to control the congestion and the transmitted data flows. This operation is done through the periodic transmission of control packets between the session participants. The RTCP main function is to provide information about the quality of the distribution of data to client application in order to diagnose communication faults. Although, there are five types of RTCP Packets, the most important are the Receiver Report (RR) and the Sender Report (SR). The RR packet carries statistical data related only to reception and is generated by the non-active participants in the media session. The SR packet carries the statistical data corresponding to reception and transmission for each active participant in the multimedia session.

IV. QOS PROPOSAL

In case of low traffic, Internet provides a Best Effort service where the packets are delivered with low delay and low losses. In case of network congestion packets are discarded without any service discrimination. The QoS concept in the Internet intends to reduce, for certain class of traffic, this uncertain packet delays and packet losses.

Since the main objective of this paper is to design and implement a solution to guarantee the quality of service in VoIP communication, it is convenient to analyze the main approaches available to provide QoS. Over provisioning is the simpler solution to provide quality of service in IP networks, however it is usually not suitable to WAN due to its bandwidth limitations. The IntServ model, defined by IETF, is based in resource reservations and uses the Resource Reservation Protocol (RSVP). DiffServ is another model defined by IETF that classifies marks and processes the packets according to a prioritization mechanism. MPLS is another mechanism that uses a tag appended to IP packets, allowing to forward traffic along the network based on traffic engineering rules.

As none of the analyzed QoS approaches is widely available in the public Internet, it was decided to design a solution able to work with the Best Effort IP network and compatible with current existing VoIP implementations.

The transmission of an audio signal through a packet switching network like Internet follows the following steps: a sequence of analog audio samples is digitalized and inserted into packets that are sent through the network to a certain destination. When the packet arrives to destination, it is decoded and reproduced. Analyzing in detail how the transmission process works, we verify that there are two key points that directly influence the quality of voice during a conversation, the coding and the transport processes.

The proposed solution aims to develop an end-to-end QoS adaptive VoIP client application. This application was called NCVoIP and is responsible to control the voice quality in a conversation. To implement the control voice quality process this application uses the R factor of the objective E-model metric. This factor represents the sum of the distortion values that are transported in the RTCP messages.

In addition to implementing the basic operation of a regular VoIP client, the NCVoIP will be composed by three QoS sub solutions that complete themselves. The first two consist in switching the codec and transport protocol in real time in a conversation and the third in using a FEC mechanism.

A. Switching Codec

As discussed above, voice codecs have different characteristics, such as transmission rate and coding delay that influence the voice quality during a conversation. For instance, if the user is connected to a low bandwidth network, it is necessary to analyze if is preferable to use the 64Kbit/s G.711 codec or the 6.3 Kbit/s G.723 codec, or any other. The answer to the question “What is the best codec in a certain situation?” represents the tradeoff between the available bandwidth, the delay and the expected voice quality. Due to the IP network characteristics it is virtually impossible to define in advance the most appropriate codec for that call.

This first part of the solution intends to automate the discovery process of the more appropriate codec to use in specific network conditions.

The process of choosing the best codec for a particular application is not an easy task because it is necessary to consider several parameters that influence the quality of the voice during a conversation. Due to the IP network characteristics it is virtually impossible to define in advance the most appropriate codec for an application.

The strategy adopted to determine the more appropriate codec to use in a given moment, is based in an iterative process that dynamically switches the audio codec when the voice quality does not correspond to the minimum requirements. To determine the codec to use when the application detects that the voice quality is low, we defined an available ordered transmission rate codec list, because as we discussed earlier, codecs with higher transmission rates have better quality. The available ordered transmission rate codec list is illustrated in Fig. 5.

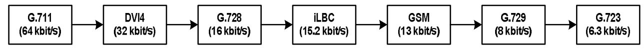

Figure 5. Available ordered transmission rate codec list

As observed in Fig. 6, when a call is established between two NCVoIP applications they use the higher transmission rate codec available, usually the G.711 codec. After establishing the call, both user applications initiate their QoS control system to evaluate the call bidirectional voice quality. When the client application detects a low voice quality, it sends a request to the other application to update the VoIP call parameters and exchange the G.711 voice codec with a lower transmission rate codec. According to Fig. 5, the next codec would be the DVI4. This process repeats itself until there is no more codecs in the available transmission rate codec list.

The codec switching process must be completely transparent to the user, in other words, the user should not realize that the codec has exchanged, he should only notice that the quality of voice has increased.

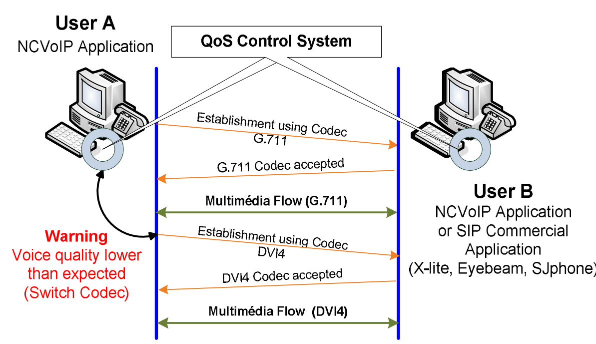

Figure 6. Real time codec switching during a VoIP call

B. Switching Transport Protocol

The end-to-end voice transport represents one of the most relevant factors to guarantee the expected QoS in VoIP communications. This way, it is necessary to define which transport protocols (UDP or TCP), fits better concerning voice quality in specific scenarios. Considering a VoIP call in a congested network, we will probably have high delay and packet losses during the call. If we are using UDP as the transport protocol, then the packet losses will be the main responsible for the quality degradation. Otherwise, if TCP is used the higher delay will be the main factor responsible for the quality reduction. This type of decision represents the tradeoff between the delay and the packet loss.

This second part of the proposed solution intends to automate the choice process of the transport protocol to use in order to optimize voice quality.

The transport protocol selection is related with the QoS parameter that we decide to minimize: delay or packet loss. In a conversation the occasional loss of one or two voice packets; although not desired, does not have a strong impact on the audio quality. The use of TCP in this case would add a considerable delay, inadmissible in media applications. Nevertheless, UDP will not always be the best option to carry voice over IP, namely in presence of a high level packet loss. In this case it is preferable to use TCP to eliminate the packet losses, at the cost of increasing delay. This tradeoff between packet loss and delay is essential to achieve a good level of quality in a conversation.

The strategy adopted to determine which transport protocol is more appropriate to carry voice in IP network, requires a permanent control of the packet loss and delay during the VoIP call. The maximum tolerable delay considered in the proposed QoS algorithm is between 100 and 200 milliseconds and the PER between 1 and 2 percent. UDP is the NCVoIP default transport protocol.

As observed in Fig. 7, when a call is established between two NCVoIP applications they use the default transport protocol, in other words UDP and both user applications initiate their monitoring system to control the call delay and the packet error rate. When the client application detects a packet error rate higher than the pre-defined, it sends a request to the other application to start using TCP.

Later if the client application detects that the call delay is higher than 200 milliseconds, it requests again the exchange of the transport protocol to UDP.

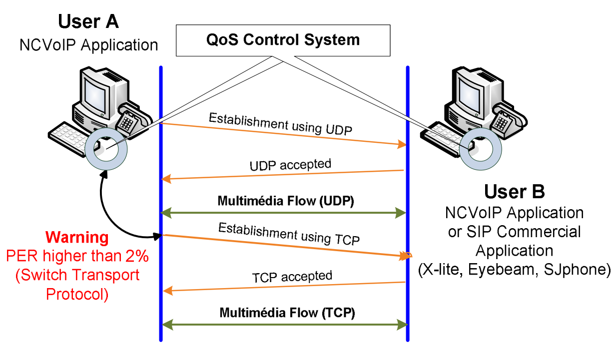

Figure 7. Real time Transport Protocol switching during a VoIP call

C. Forward Error Correction

Wireless channels, typically, have high error rates caused by attenuation, dispersion or interference from other sources. Consequently in these environments it is necessary to use an error correction technique to guarantee high voice quality. FEC emerges as the best solution to solve the problem of lost packets.

However, despite this technique is included as one of the proposed solutions in this project, it is not implemented in this version of the NCVoIP application, because further work is required to choose the most appropriate FEC mechanism and the redundant data block size.

D. Proposed Quality of Service Algorithm

The proposed QoS algorithm comprises the combination of the first two components of the solution introduced in this chapter. This algorithm is represented in Fig. 8, and seeks to discover and use automatically the more appropriate VoIP call parameters in a defined moment.

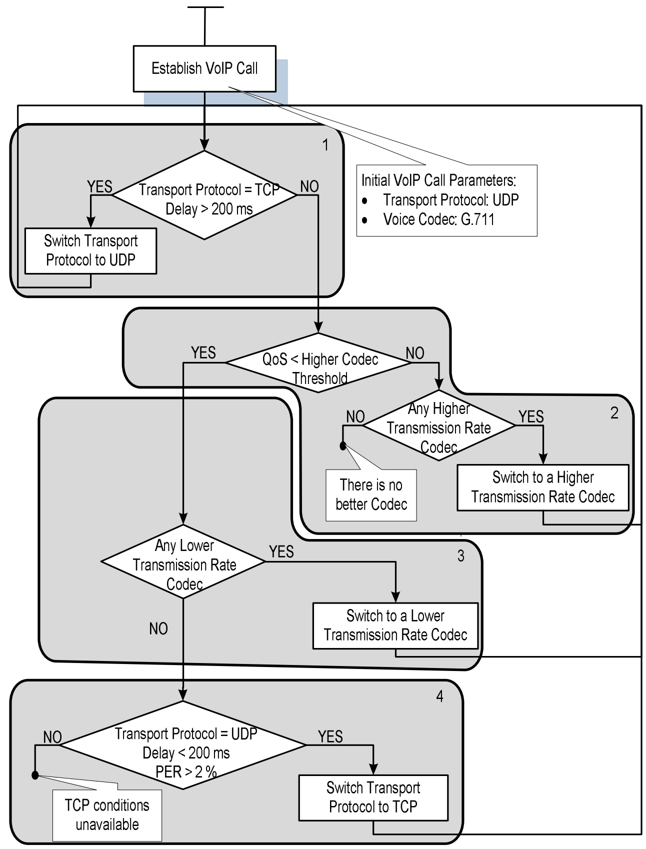

Figure 8. Quality of Service Proposed Algorithm

When a voice call is established, it starts using the UDP protocol and the highest transmission rate codec available, usually G.711. The NCVoIP application starts to regularly monitor and analyze the quality of the voice in order to assure the maximum VoIP quality. When NCVoIP detects that the quality of voice is outside the predefined values of each codec, it verifies if it is larger than the upper threshold of the Codec used, meaning that it is possible to change to a higher transmission rate Codec, if available. This process is represented in the module 2 of Fig. 8.

When the voice quality is lower than the lower threshold, the application NCVoIP checks if a lower transmission rate Codec is available and requests a Codec switch. This process takes place whenever the voice quality is lower than the expected and while there is a lower transmission rate Coded available, as exemplified in module 3 of Fig. 8.

In case there is no lower transmission rate Codec available, the delay is less than 200 ms, the PER is greater than 2% and the transport protocol used is the UDP, NCVoIP requests to switch the transport protocol to TCP. This process is represented in the module 4 of Fig. 8. After changing to TCP protocol, it becomes necessary to periodically verify if the delay is greater than 200 ms and if the PER is less than 2%. In this case, it is necessary to switch again the transport protocol to UDP, as explained in module 1 of Fig. 8.

To avoid frequent changes of transport protocol that could cause the transmission of a high number of signaling messages, a timer is introduced after each change of transport protocol.

E. Examples of application of the proposed algorithm

Fig. 9 illustrates the application of the proposed algorithm to codec switching, using the contours of user satisfaction presented in G.109 - Amendment 1 [16]. The terminal is initially using codec G.711 at 64 Kbit/s with low delay and low losses (point A in the figure).

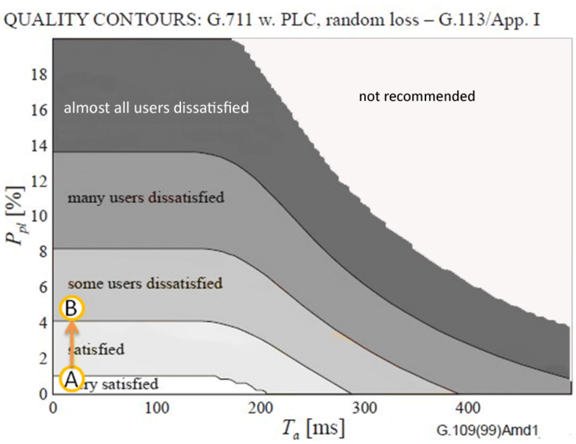

Figure 9. Initial state of the terminal with G.711 codec

If in a certain moment in time there is congestion in the network, it could cause severe packet losses in the VoIP connection (point B in the figure), degrading the VoIP quality the region “some users dissatisfied”.

As a consequence of this quality degradation, the proposed algorithm switches to the next codec with lower bitrate in order to decrease the congestion. This codec change could be done together with the activation of a FEC mechanism, in order to guarantee that the packet losses will be much lower than before.

In the example presented the new codec available is G.729A at 8 Kbit/s, which due to its much lower bitrate and the use of FEC will result in a much lower packet losses, but with higher end-to-end delay (point C in Fig. 10).

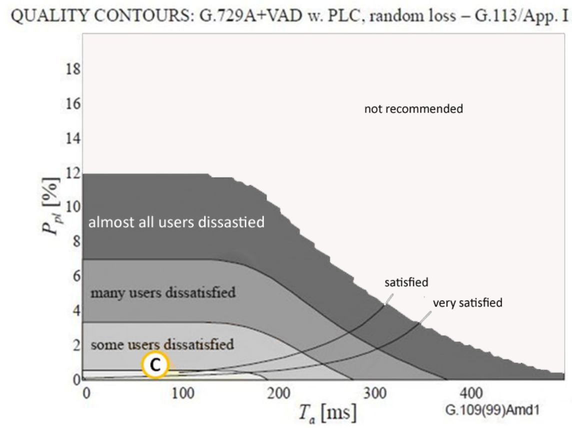

Figure 10. Terminal after switching to G.729A codec

As can be observed in Fig. 10, the terminal returned to the “very satisfied” region, what means that its quality is again very good, as desired.

Fig. 11 illustrates the application of the proposed algorithm to protocol switching. The terminal is initially at point A in this figure, as in the previous case, with low delay and low packet losses.

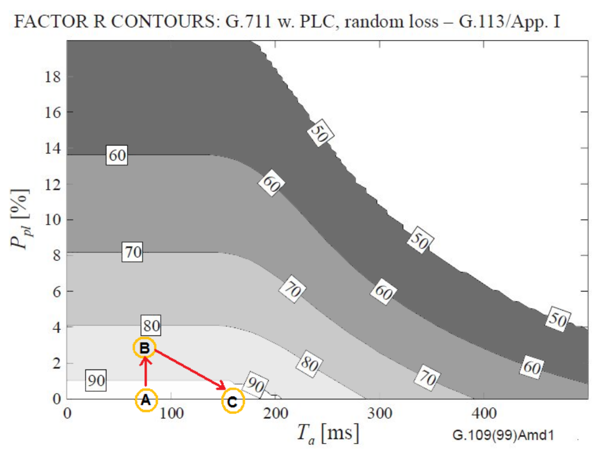

Figure 11. Transport Protocol switching sequence

If in one moment there is a sudden increase of packet losses, for instance due to noise or interference typical of wireless environments, the VoIP connection suffers high packet losses (point B in the figure).

The proposed algorithm detects this and considering that there is no other codec to switch, decides to change the protocol from UDP to TCP, in order to achieve a reliable connection, without losses. This change implies that the end-to-end delay will increase due to the occasional retransmission that happen with the TCP protocol, however if the Round Trip Time (RTT) of the TCP connection is not very high, the voice quality will still very good, since the new terminal operating point (C) will again be inside the region where the R factor is higher than 90, as can be observed in Fig. 11.

V. NCVoIP Implementation

Since the main objective in developing a VoIP client application was to implement and test the QoS proposed solution, it was decided to use the open source SipCommunicator Java application. This application uses the Jain-SIP v1.2 library to implement the SIP protocol and the Java Media Framework v2.1.1 library to implement the RTP and RTCP protocols.

Fig. 12 shows the architecture of the SipCommunicator client application.

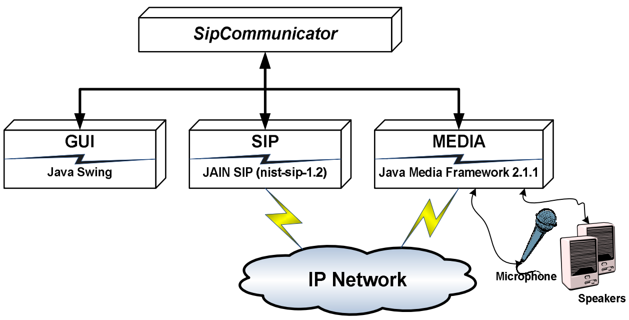

Figure 12. SipCommunicator Architecture

A. NCVoIP Components

NCVoIP architecture is shown in Fig. 13 and is structured in the five following components:

- GuiManager is responsible for the NCVoIP graphical interface presentation and its interaction with the user. This module communicates with the SipManager module when the user solicited operations that are directly related to the establishment, termination or updating of the characteristics of the call VoIP.

- SipManager, implements and manages the SIP state machine. This component creates all SIP messages that allow establishing, updating and closing a VoIP call. SipManager communicates with the MediaManager module to start, maintain and terminate the multimedia flow during a VoIP call. This module also interacts with the DoQoS module when it is necessary to change a call parameter or when it receives a request to update the characteristics of the call.
  The operation to update the parameters of a VoIP call was completely implemented from scratch, since it was not included in the SIP base application. This operation was implemented using the SIP method UPDATE, defined by IETF in RFC 3311 [17]. This method allows a participant to submit a request to update the characteristics of a VoIP call in a session.
- MediaManager, generates, transmits and receives RTP and RTCP packets. This module is also responsible for capturing and playing the multimedia data.
  Upon receiving a Sender Report or Receiver Report packet, MediaManager communicates with EvalQoS in order to calculate the R factor value. MediaManager also receives indications from SipManager to start, maintain and terminate the multimedia flow during a VoIP call.

- EvalQoS is responsible for calculating periodically the E-Model R factor value based on the equations presented earlier and the values carried in RTCP packets. After calculating the R factor, this module communicates with DoQoS to evaluate and decide if it is necessary to make any change in the VoIP call parameters to increase the voice quality.

- DoQoS implements the proposed algorithm and is responsible for evaluating and deciding whether it is necessary to change any call parameter and specifies the modifications. In case it is necessary to change any call parameter, DoQoS invokes SipManager to initiate the call update process.

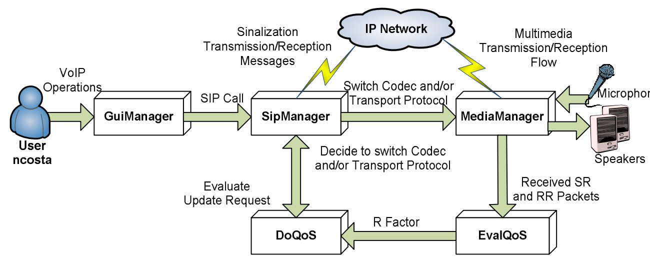

Figure 13. NCVoIP Architecture

VI. NCVoIP Voice Quality Measurements

In this chapter we present the scenarios to test, analyze and demonstrate the feasibility of the proposed solution.

The voice codecs provided by the Java Media Framework library are G.723.1, IMA/DVI4, ULAW and GSM. In order to define the order and the R factor limits associated with these codecs, it was performed a set of tests without any restriction, to measure the R factor for each codec. The duration of each test was approximately 300 seconds. Table 3 shows the results of the set of tests performed.

TABLE III. Results of all tests performed

Codec	Transmission Rate [Kbit/s]	Mean Jitter [ms]	Mean Delay [ms]	PER [%/s]	Mean R Factor
G.711	64	19,21	74,31	0	91,23
DVI4	32	19,09	74,24	0	89,00
GSM	12,2	18,60	73,73	0	86,26
G.723	6,3	17,20	72,20	0	76,55

Analyzing the values of table 3, it appears that as discussed earlier, codecs with higher transmission rate have the highest R factor and so gradually. Therefore, the list of available codecs was ordered by its transmission rate, as shown in Fig. 14.

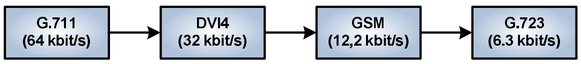

Figure 14. Ordered available JMF voice codec list

Looking again to the values of table 3, we can also determine the upper R factor limit of each codec, which will be used by the NCVoIP application to switch codec. Table 4 shows the R factor limits for each codec. This table does not include the G.711 codec, as it is the highest available voice quality codec and has no upper limit.

TABLE IV. R FACTOR CODEC HIGHER LIMITS

Codec	Higher Limit
DVI4	89
GSM	86
G.723	76

To set the lower R factor limits it was necessary to perform a set of tests using a scenario with high error rate such as the wireless network. To gather these conditions we implemented, in the NVoIP application, a mechanism to simulate packet losses in a controlled manner. It was also used the Iperf application to cause congestion and delays on the network. Therefore, using the Iperf application we were able to overload the wireless IEEE 802.11g (54 Mbit/s) with 10 Mbit/s UDP traffic.

Table 5 shows the lower R factor limits for each codec. This table does not include the G.723 codec, because it is the lower voice quality codec available.

TABLE V. R FACTOR CODEC LOWER LIMITS

Codec	Lower Limit
G.711	86
DVI4	84
GSM	74

Based on the results of table 4 and table 5 we created the table 6. This table presents the threshold limit values used for each available codec.

TABLE VI. CODEC THRESHOLD LIMITS

CODEC	Upper Limit	Lower Limit
G.711	86	-
DVI4	84	89
GSM	74	86
G.723	-	76

A. Bandwidth Limitation Test Scenario

This scenario, illustrated in Fig. 15, demonstrates how the NCVoIP voice quality behaves when there is little available bandwidth to establish a VoIP call.

To precisely limit the available bandwidth, we used the NetLimiter 2 Pro application.

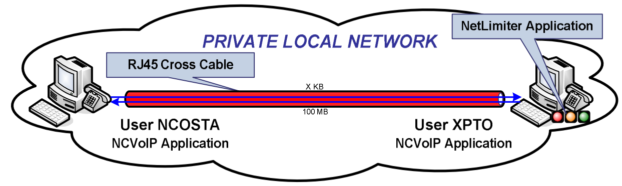

Figure 15. Bandwidth limitation test scenario

Fig. 15, illustrates the defined test scenario, which is composed by two users (NCOSTA and XPTO) that have installed the NCVoIP application to establish a VoIP call. User XPTO has also installed the NetLimiter application to limit the NCVoIP incoming bandwidth.

For each NCVoIP voice Codec available we produce two voice quality measuring tests, the first without any bandwidth limitation and the second with a pre-defined bandwidth. Every 200 seconds, user NCOSTA manually switches the voice Codec and observes its influence in the voice quality.

1) *Scenario 1*

In scenario 1, we defined an available bandwidth equal to 99 % of the Codec transmission rate.

TABLE VII. BANDWIDTH LIMITATION TEST SCENARIO 1

Codec	Bandwidth [Bytes]	Interval Min.	Interval Max.	Mean Delay	Mean PER	Mean R Factor
G.711	-	0	209	72.1	0.0	91.7
G.711	8110	209	399	133.6	0.4	88.7
DVI4	-	399	609	70.7	0.0	89.2
DVI4	4055	609	825	101.8	0.2	87.1
GSM	-	825	1053	69.7	0.0	86.8
GSM	1545	1053	1298	129.7	0.3	82.3
G.723	-	1298	1597	70.6	0.0	76.7
G.723	797	1597	1902	116.4	0.2	75.1

Table 7 demonstrates, as expected, that due to a packet loss increase, when we limit to 99% the available bandwidth, the R factor decreases considerably. Observing the PER values, we verify that limiting the bandwidth result in packet losses that directly influence the voice quality.

Analyzing the voice quality that resulted of the proposed QoS algorithm, we verify that in some situations it corresponds to a voice quality increase. For example, if we consider the switching process from Codec G.711 to codec DVI4, we observe that it resulted in an increase of the mean R factor from 88.76 to 89.24. On the other hand, if we consider the switching process from Codec DVI4 to GSM, we verify that it does not result in a voice quality increase, but the values are pretty close.

B. Network Congestion Test Scenario

The test scenario defined to demonstrate the voice quality achieved when two users (NCOSTA and XPTO) establish a VoIP call in a congested network is represented in Fig. 16.

User A and XPTO are connected to a 100 Mbit/s Ethernet router and user B and NCOSTA are connected to a 10 Mbit/s hub.

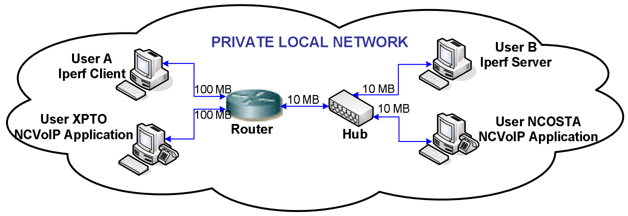

Figure 16. Network congestion test scenario

In order to simulate a network with high level of congestion we used the Iperf application to generate UDP traffic between the router and the hub. This scenario only uses the voice Codec G.711 and we switch manually the transport protocol, approximately every 300 seconds.

1) Scenario 1 – Reference Model

This first scenario represents the reference model scenario that will be used to compare the voice quality resulted by UDP and TCP protocols in the following scenarios. In this scenario, the Iperf traffic was disabled.

TABLE VIII. SCENARIO 1 TRANSPORT PROTOCOL SWITCHING SEQUENCE

Protocol	Interval		Mean	Mean	Mean
	Min.	Max.	Delay	PER	R Factor
UDP	0	294	73	0.0	91.7
TCP	294	603	121	0.0	90.6

Analyzing the values in table 8, we verify that the mean R factor in UDP is higher that the mean R factor in TCP, as expected.

2) Scenario 2

In this scenario we enabled the Iperf traffic and defined the Iperf client to send UDP datagram packets with a transmission rate of 8 Mbit/s.

TABLE IX. SCENARIO 2 TRANSPORT PROTOCOL SWITCHING SEQUENCE

Protocol	Interval		Mean	Mean	Mean
	Min.	Max.	Delay	PER	R Factor
UDP	0	295	92	0.7	89.0
TCP	295	621	130	0.0	90.2

Comparing the values in table 8 and table 9, we verify that, despite slight, the initial situation has been reversed and the TCP mean R factor is higher that UDP mean R factor.

Analyzing the values used to calculate the UDP and TCP mean R factors we verified that the UDP values are inconstant, while TCP values are constant.

3) Scenario 3

In this scenario we opted to further saturate the 10 Mbit/s connection by defining the Iperf client to send UPD datagram packets with a transmission rate of 9.5 Mbit/s.

TABLE X. SCENARIO 3 TRANSPORT PROTOCOL SWITCHING SEQUENCE

Protocol	Interval		Mean	Mean	Mean
	Min.	Max.	Delay	PER	R Factor
UDP	0	322	221	17.2	46.0
TCP	322	632	384	0.0	71.3

This scenario highlights even more the fact that in this situation TCP voice quality is better than UDP voice quality.

Analyzing the values in table 10, we verify that despite TCP introduces a significant delay, it is able to recover the loss packets, resulting in approximately 24 points higher than UDP.

VII. FEC EVALUATION SIMULATIONS

Since FEC was one of the solutions proposed in this paper to restore automatically the quality of voice in a VoIP call, we have done a detailed analysis to define which FEC mechanism and redundant information is the most appropriate in VoIP communications.

A. *Simulation Environment*

The simulation environment, represented in Fig. 17, was developed with the Simple Simulation Kernel (SSK) [18] simulator and it consists of five main blocks: TCP traffic generators, VoIP traffic generator, IP network, TCP traffic destination and VoIP traffic destination.

TCP generators simulate the best-effort traffic, VoIP generator, generate a voice G.711 packet every 20 ms. The IP network consists of two routers that have multiple queues, allowing simulating priorities and delays in packet transmission. The TCP and VoIP destinations are responsible to receive the correspondent traffic.

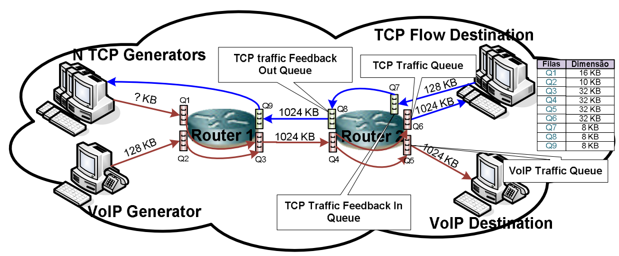

Figure 17. FEC simulation environment

B. *Simulations Performed*

Since we want to study the behavior of VoIP traffic in the developed simulation environment, we identified three parameters that can directly affect the performance of VoIP traffic along the path. These parameters are: size of queue Q3, number of TCP traffic generators and their respective transmission rate. The possible transmission rates are: 128 Kbit/s, 256 Kbit/s, 512 Kbit/s and 1024 Kbit/s. The TCP generators are: 1, 2, 4, 8 and 10. Each simulation has duration of 400 seconds.

In order to evaluate more accurately the influence of FEC in VoIP traffic, first we have done a set of simulations without applying the FEC technique and then repeated the simulation with FEC. Examining the packet loss data obtained in the first phase of the simulations, we verified that the percentage of an isolated error occurrence is between 86% and 92%. Therefore, we have chosen to use a simple parity correction code FEC mechanism.

Since one of the objectives proposed in this chapter is to find the more appropriate packet block size to generate a FEC packet, we have considered the following values: 1, 2, 3, 4, 6, 8, 10, 12, 14 and 16.

Fig. 18 shows the way a FEC packet is generated from a four VoIP packet block.

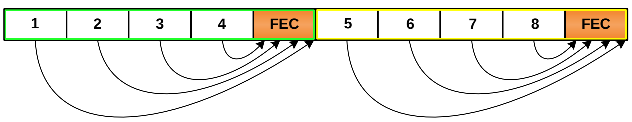

Figure 18. FEC packet generation

Since the second phase uses the first phase delay and PER values, it was necessary to correct those values due to the introduction of redundant packets in the network. The methodologies used to calculate the average values of delay and PER in this phase are explained above.

1) Mean Delay and Id Parameter Calculation

The delay value is calculated in function of the packet block size used to generate the FEC packet. The bigger is the block size, the higher is the delay, because the receiver needs to wait for the reception of all the packets of the block until it gets the FEC packet to recover and play the missing packet.

Fig. 19 illustrates how to calculate the absolute delay when using the FEC technique to provide redundancy and be able to recover lost packets or packets with errors.

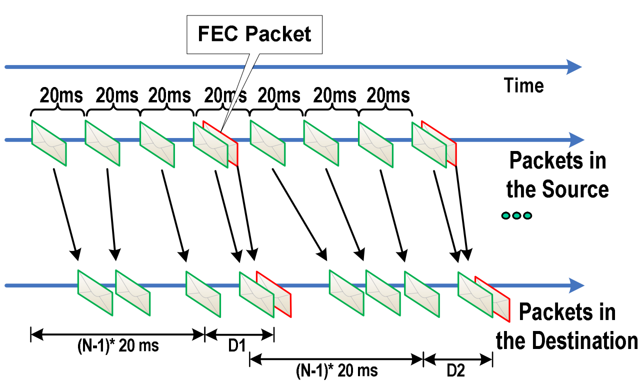

Figure 19. Absolute delay calculation

For each simulation, we grouped into blocks of N, the delay value of the VoIP packets and applied to equation 6 to obtain the new delay value.

$$newDelay = ((N-1)\times 20\,ms) + D \qquad (6)$$

N represents the block size and D the delay of the last block packet. After calculated the absolute delay we used expression 3 to calculate the new I_d value.

2) Mean PER and Ie Parameter Calculation

The PER calculation process consists in grouping in blocks of N the first phase PER values and verify if there is more than one error. Using the XOR operation we can recover one packet error per block and decrease the PER value. We define PERGain as shown in equation 7.

$$PERGain = \frac{number\ of\ packets\ recovered\ per\ second}{number\ of\ packets\ lost\ per\ second} \qquad (7)$$

Based on the value of the PERGain and the PER average value in the first set of simulations, it is possible to calculate the new PER average value obtained with the use of a FEC block, as illustrated in equation 8.

$$newPER = (1 - PERGain) \times PER \qquad (8)$$

The I_e parameter results from the sum of I_{eloss} and I_{e0}. I_{eloss} corresponds to the equipment impairments as a function of PER. This component is directly obtained when the PER value is represented in the Fig. 4 graphic. Otherwise, I_{eloss} value is obtained through the line equation that passes between the PER adjacent points.

I_{e0} represents the overhead resulted by the introduction of the FEC packet on the IP network. To calculate this component value, we had to obtain the transmission rate correspondent to send the N + 1 packet and then use the Fig. 4 graphic.

3) VoIP over UDP and Q3 with 16 Kbyte

This first simulation group analyzes the FEC impact in VoIP traffic when router 1 has a Q3 queue of 16 Kbyte.

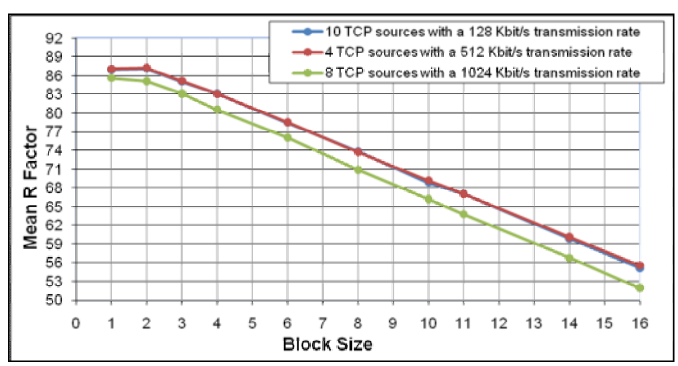

Figure 20. Simulation “VoIP sover UDP and Q3 with 16 Kbyte” graphic

According to Fig. 20, in simulations of “10 TCP generators with a transmission rate of 128 Kbit/s” and “4 generators with a transmission rate of 512 Kbit/s”, it is only efficient to use a FEC block lower than 4. In both simulations the highest R factor value happens when we have a two packet block.

4) VoIP over UDP and Q3 with 8 Kbyte

This second simulation group analyzes the FEC impact in VoIP traffic when router 1 has a Q3 queue of 8 Kbyte, as illustrated in Fig. 21.

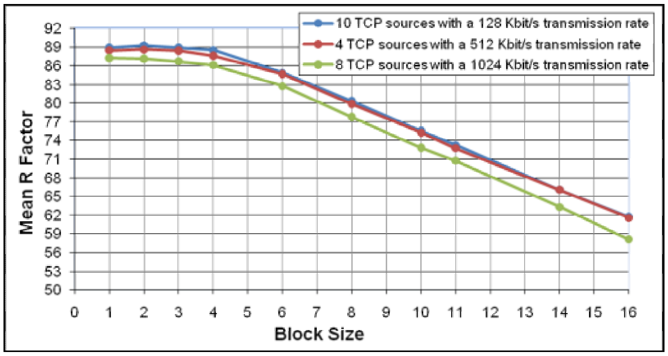

Figure 21. Simulation "VoIP over UDP and Q3 with 8 Kbyte" graphic

Analyzing the simulations "10 TCP generators with a transmission rate of 128 Kbit/s" and "4 TCP generators with transmission rate of 512 Kbit/s", we verify that it is only efficient to use a FEC block lower than 6. In simulation "8 TCP generators with transmission rate of 1024 Kbit/s", it is only efficient when the FEC block is lower than 8 packets.

Simulation "10 TCP generators with transmission rate of 128 Kbit/s" presents the highest R factor value when a 2 or 3 FEC block is used. Simulation "4 TCP generators with transmission rate of 512 Kbit/s" has the higher R factor value when a 2 packet FEC block is used.

VIII. CONCLUSION

This paper proposes an adaptive QoS strategy, without changing the IP network, to solve the problem of momentary drop of voice quality in a conversation and improve the overall quality of service.

To demonstrate the voice quality improvement resulting from this proposal, we developed the NCVoIP application. Based on this application, a set of voice quality measuring tests were performed. We verified that the proposed algorithm improved the quality in some scenarios, while in others there was no voice quality increase. However, based on these experimental tests, we proved that switching the voice Codec when the bandwidth is below the transmission rate of the used Codec and using TCP to encapsulate the RTP packets, when a congestion network exists, corresponds to a significant voice quality improvement. The switching of the CODEC during the communication could cause a short gap in the conversation, due to buffer re-initialization. Special care should be taken to avoid that switching of transport protocol cause losses in the audio stream.

In order to study the influence of FEC in VoIP traffic, we developed a simulation model and perform a set of simulations. Based on these simulations we verified that it only compensate, in terms of voice quality, to use FEC when the PER is higher than 1% and that the most appropriate FEC block is two.

REFERENCES

[1] Nelson Costa, Mário Nunes, "Adaptive Quality of Service in Voice over IP Communications", Proceedings of the 2009 Fifth International Conference on Networking and Services (ICNS 2009), IEEE Computer Society, Valencia, Spain, Abril 2009, pp. 19-24.

[2] J. Wroclawski, "The use of RSVP with IETF Integrated Services", RFC 2210, Internet Engineering Task Force, September 1997.

[3] S. Blake et al, "An Architecture for the Differentiated Services", RFC2475, Internet Engineering Task Force, December 1998.

[4] E. Rosen *et al*., "Multiprotocol Label Switching Architecture", IETF RFC 3031, January 2001.

[5] M. Narbutt, A. Kelly, "Adaptive VoIP Play-out Scheduling: Assessing User Satisfaction", IEEE, Vol. 9 Nº4, July 2005.

[6] Liang-Yi Huang Yung-Mu Chen et. Al, "Adaptive VoIP Service QoS Control based on Perceptual Speech Quality", The 9th International Conference on Advanced Communication Technology, February 2007, Volume 2, pp. 885-890.

[7] Wojciech Mazurczyk1, Zbigniew Kotulski, "Adaptive VoIP with Audio Watermarking for Improved Call Quality and Security", Journal of Information Assurance and Security, September 2007, pp. 226-234.

[8] See Leng Ng, Simon Hoh, Devinger Singh, "Effectiveness of adaptive codec switching VoIP application over heterogeneous networks", 2nd International Conference on Mobile Technology, Applications and Systems, November 2005, pp.1-7.

[9] ITU-T G.107, "The E-Model, a computational model for use in transmission planning", March 2003.

[10] A. D. Clark, "Modeling the Effects of Burst Packet Loss and Regency on Subjective Voice Quality. IP – Telephony Workshop", 2001.

[11] L.C.G. Lustosa, L.S.G. Carvalho, P.H.A. Rodrigues, E.S. Mota, "Using the E-Model to evaluate the quality of speech in Voice over IP communication systems", 2004.

[12] ITU-T G.113 Appendix I, "Provisional planning values for the equipment impairment factor Ie", October 2001.

[13] Ling-Jyh Chen, Tony Sun, M. Y. Sanadidi, Mario Gerla, "Improving Wireless Link Throughput via Interleaved FEC", UCLA, USA, 2004.

[14] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, IETF, June 2002.

[15] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, IETF, July 2003.

[16] ITU-T G.109 Amendment 1, "The E-model based quality contours for predicting speech transmission quality and user satisfaction from time-varying transmission impairments", January 2007.

[17] J. Rosenberg, "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, IETF, September 2002

[18] J. Azinhais, J. Jesus, "SSK – A Simple Simulation Kernel for UNIX & MS-DOS environments", INESC, 1998