A New Adaptive Redundancy Control Algorithm For VoIP Applications

Haytham Assem, David Malone
Hamilton Institute, National University of Ireland
Maynooth, Ireland
Email: {Hitham.Salama.2012, David.Malone}@nuim.ie

Jonathan Dunne, Pat O'Sullivan
Systems and Performance Engineering,
IBM Dublin, Software Lab, Ireland
Email: {jonathan_dunne, patosullivan}@ie.ibm.com

Abstract— **Packet loss is one of the most important factors in degrading Voice over IP (VoIP) perceived call quality. Forward Error Correction (FEC) is a powerful technique for transmitting audio streams over the IP network to decrease the effect of packet loss. Although these method reduces the effect of packet loss, it increases the bandwidth and delay in order to recover from the lost packets. In this paper, we propose a new adaptive FEC mechanism for voice calls based on the generated codewords from a Reed-Solomon (RS) encoder. This mechanism chooses the optimum RS code from a family of codes to improve the conversational call quality. Our proposed mechanism is able to switch between different codes during the call to account for the variation of the network conditions including packet loss and delay. We have deduced the proposed algorithm by performing subjective mean opinion score (MOS) testing based on an interactive assessment tests. We show that our adaptive algorithm outperforms fixed RS codes under highly varying network conditions.**

Keywords—VoIP; FEC; Reed-Solomon; Call quality; MOS

I. Introduction

Voice over Internet Protocol (VoIP) applications have become important applications and are expected to carry more and more voice traffic over TCP/IP networks. One of the challenges in VoIP networks is measuring the voice quality accurately and efficiently. ITU-T provides two test methods subjective and objective testing. Subjective testing was considered by the earliest studies evaluating the speech quality by giving Mean Opinion Scores (MOS). ITU-T Rec. P.800 [1] presents the MOS test procedures; MOS test is a subjective listening test where it usually involves 12-24 participants who individually listen to an audio stream of several seconds and rate the audio quality on the scale of 1 (Poor) to 5 (Excellent). Although subjective testing using MOS is time consuming, expensive and does not allow real time measurements, it is still the reference and the most accurate measurement for the perceived voice quality. In the last years, several techniques were developed for measuring MOS scores in an objective way (without human perception): PESQ [2] and E-Model [3] are some of the most popular methods for doing so. PESQ is based on the comparison of two signals to generate the MOS—a reference signal (e.g. captured at the sender) and a degraded signal (e.g. captured at the receiver). The requirement for comparison of both signals makes the approach unsuitable for live monitoring. In contrast, the E-model technique, specified in ITU-T Rec. G.107 [3], is a non-intrusive method that uses network metrics locally monitored at the sender to estimate call quality, so it can be used for live call monitoring.

Packet loss in the IP network is one of the most important factors that cause degradation in the overall voice call quality—packet loss greater than 5% has been shown to have a very detrimental effect on voice quality [4]. The importance of having low packet loss rates to sustain high perceived voice call quality led to a number of loss repair methods introduced (e.g.: FEC and low bit-rate redundancy (LBR)). It was shown previously that FEC is preferable over LBR [5]. Forward Error Correction (FEC) recovers lost packets by transmitting redundant data. FEC schemes send redundant information with the original information to recover the lost original information. Nowadays, the Reed-Solomon codes are one of the most commonly used FEC coding schemes in the VoIP applications [6]. Reed-Solomon codes are convenient technique for VoIP applications, as every RTP packet can be represented as one of the data symbols of a codeword while the parity bits will hold some redundant information based on previous packets. In general, the major drawback of using FEC is the increase in the delay because the receiver cannot start the playback after a loss until it receives the parity data. Therefore, an adaptive redundancy control algorithm must consider the level of FEC scheme applied to stream of RTP packets carefully. An optimization problem arises from here to choose an appropriate FEC scheme to be applied with the guarantee of higher perceived call quality. It is precisely the goal of this paper to solve this problem. We have observed in many current VoIP applications that the delay factor is not taken into account when adjusting the coding scheme [6]. This may result in recovering some of the packet losses at the expense of crossing the acceptable delay level. This may lead to worsen the overall call quality. Thus, we propose our redundancy control algorithm, the "APU algorithm" (Acceptable Poor Unacceptable Algorithm), which chooses the optimum RS code during the call taking into account the tradeoff between losses recovered and delay overhead added.

This paper is organized as follows. In Section II we review some of the relevant previous work. In Section III, we describe the Reed-Solomon codes and our MOS comparative analysis when using different RS codes. We propose our

The authors were supported by Science Foundation Ireland (SFI) grants 07/SK/I1216a and 08/SRC/I1403.

978-1-4799-1353-4/13/$31.00 ©2013 IEEE

subjective interactive testing results and our APU algorithm in Section IV. Section V describes our simulation results. In section VI, we conclude and summarize our paper.

II. Related Work

Jiang and Schulzrinne studied loss repair methods applied to VoIP applications. In [5, 7], they studied the relation between the packet interval and the FEC performance. They performed subjective testing to see the influence of bursts losses on the perceived quality. They stressed two main points in their work. First, FEC is preferred over LBR if the codec used is a low bit rate codec. Second, they found that G.729 with FEC generally prevails compared to other codecs (G723.1 and ILBC). In [8], Rousan and Nawarash introduced the Bandwidth Optimized Adaptive FEC (BOAFEC) approach to optimize the redundancy of the generated codewords from Reed-Solomon (RS) encoder in order to save bandwidth. This approach succeeded in saving 25% of the redundant bandwidth, which allows for more clients on same server. This scheme responds to high network losses by setting the maximum allowed amount of redundancy. Recently, Huang, Chen, Hunang and Wang in [6] have examined the amount of redundancy Skype adds to its voice streams. They show that Skype's control algorithm does not take into account the individual codec and bursts loss factors. This leads them to derive an optimal redundancy control policy for a desired quality under certain network conditions for G.711 and G.729 codecs. They derived the equivalent redundancy ratio for each PESQ score using two different RS codes.

In contrast with these works, we focus on developing a redundancy control algorithm addressing the tradeoff between the delay overhead increased and packet loss recovered as a result of using certain RS code from the human perception point of view. We have studied this tradeoff using a subjective testing method. One of the main challenges in the previous work is that deriving the redundancy ratio was based on the PESQ MOS which does not take into account the delay overhead factor in the call quality whilst using FEC increases the delay overhead affecting the call quality at the end user. Thus, we have addressed this challenge and we propose the "APU Algorithm". Our algorithm chooses between different RS codes during the call by comparing the current call quality with the expected call quality if a certain RS code is used. Our algorithm differs from those proposed before as it takes into account the delay factor when using certain RS code.

III. MOS of Reed-Solomon Codes

One of the most used FEC codes in VoIP applications are Reed-Solomon (RS) codes. A RS code is defined by parameters (n, k); n is the total number of packets and n-k indicates the amount of parity added as seen in Fig. 1. RS (n, k) code can recover all losses in the same FEC block if and only if k out of n packets are received successfully.

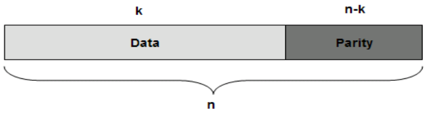


Fig. 1. RS codeword.

In this paper, we will focus on 3 main codes: RS (2, 1), RS (3, 2) and RS (4, 3). Under RS (2, 1) coding, the voice packet is lost during transmission if and only if the next packet that carries information about it is dropped as well. While, under RS (3, 2) and RS (4, 3), the voice packet is lost if and only if the next 2 and 3 packets are lost respectively.

We established VoIP calls under different network conditions using 3 different RS codes. We use a single code per call. We tested the effect of using fixed RS codes on the call quality compared when using no codes. An audio stream of 10 seconds was played with non-audible gaps in order to have accurate results when emulating losses.

We measured the voice call quality using PESQ algorithm using 2 different codecs (G.711 and G.723.1) under packet loss rate ranges from 0-20% with burst ratio (Ratio of the average length of consecutive losses under burst losses to that under random loss) equals to 1, 2 and 3. Fig. 2 shows sample of our results, where the x-axis represents the packet loss while the y-axis represents the PESQ MOS.

We stress two points from the graphs seen in Fig. 2: First, the performance of the RS code is highly dependent on the codec used, packet loss rate and burst ratio; thus these QoS factors must be taken into account when developing a redundancy control algorithm. Second, although the MOS which indicates the QoE at the end user, shows that RS(4,3) prevails compared to other RS codes, this may change when adding the delay overhead of the different codes when measuring the conversational MOS during the call.

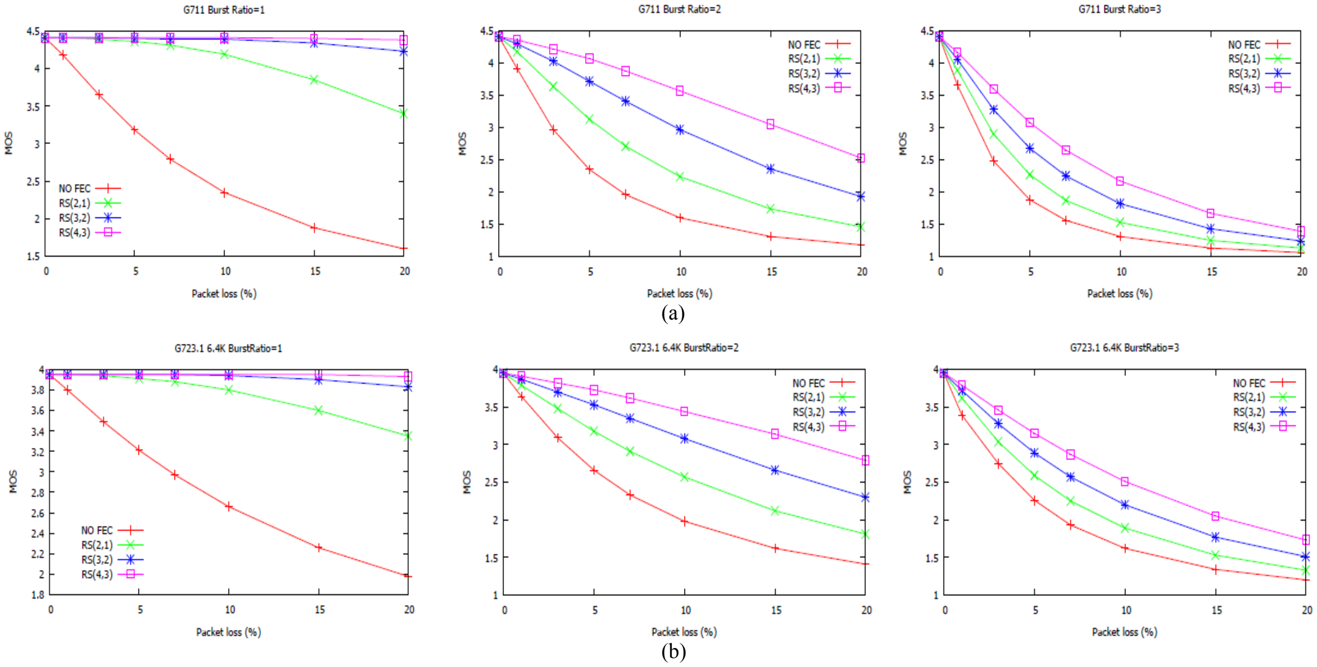
IV. APU Algorithm

In this section, we propose our adaptive FEC algorithm taking into account the packet loss, burst ratio, codec used and the overall delay. We aggregate the first 3 factors in the MOS while the fourth factor (delay) was studied separately and combined with the MOS to derive a single metric called "APU score". We use it to rate the conversational call quality during the on-going call. The main advantage of our algorithm is that it takes into account the delay overhead as a result of using different RS codes.

We take the decision to use certain RS code based on this metric (APU score) derived from subjective testing. We divide this section into 3 parts. First, we introduce the APU (Acceptable, Poor and Unacceptable) levels of the MOS and delay factor in order to identify all test cases required for subjective testing. Second, we show the results of our subjective testing. Third, we propose our adaptive algorithm.

A. APU Model for MOS and one-way delay

ITU-T G.107 Recommendation [3] provides the satisfaction level corresponding to a measured MOS. Since, we need to derive an algorithm dependent on the codec used, thus we cannot rely on such relation. For instance, the maximum achievable MOS of G.711 codec is 4.41 indicating, "Very satisfied" whilst the G.726 16K has a maximum MOS of 2.74 indicating, "Nearly all users are dissatisfied". We want to assess the codecs subject to the performance of each codec. So in the prior example, 2.74 should be satisfying if the G.726 16K is used and would be unsatisfying if G711 codec is used.



(a)

(b)

Fig. 2. Different RS codes applied to (a) G.711 and (b) G.723.1 6.4k codecs with different loss burst ratio.

In another words, we need to have a relative relation so that we can assess the call quality based on the codec used in order to take an appropriate decision whether it is required to improve the call quality or not. In the prior example, it might be a possible target to improve the call quality using G711 codec because the codec itself can attain higher call quality unlike the case using G726 16K. To solve this problem, we have measured the peak performance of different codecs under no network impairments using PESQ algorithm and we divided this into 3 equal bands in order to derive the APU model for the MOS based on the codec used (See Table I).

The effect of the delay is relatively significant on VoIP call quality. Based on ITU-T Recommendation G.114 [9] for one-way delay transmission time, it is straightforward to partition delay ranges to reflect our APU model as seen in Fig. 3.

TABLE I. APU MODEL FOR THE MOS OF SOME CODECS

Codec	Acceptable	Poor	Unacceptable
G711	3.28-4.41	2.14-3.28	1-2.14
G723 5.3k	2.86-3.79	1.93-2.86	1-1.93
G723 6.3k	2.96-3.95	1.98-2.96	1-1.98
G726 16k	2.16-2.74	1.58-2.16	1-1.58
G726 32k	3.04-4.07	2.02-3.04	1-2.02
G726 40k	3.16-4.24	2.08-3.16	1-2.08
G729	3.26-4.13	2.13-3.26	1-2.13
GSM FR	2.64-3.46	1.82-2.64	1-1.82
G729 A	3.06-4.10	2.03-3.06	1-2.03

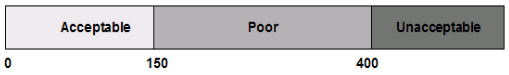


Fig. 3. APU delay model.

In this paper, we will use <$d_{network}$, $MOS_{network}$> notation to define the state of the VoIP call, where $d_{network}$ indicates the overall one-way delay while the $MOS_{network}$ indicates the call quality taking into account the codec used and network losses.

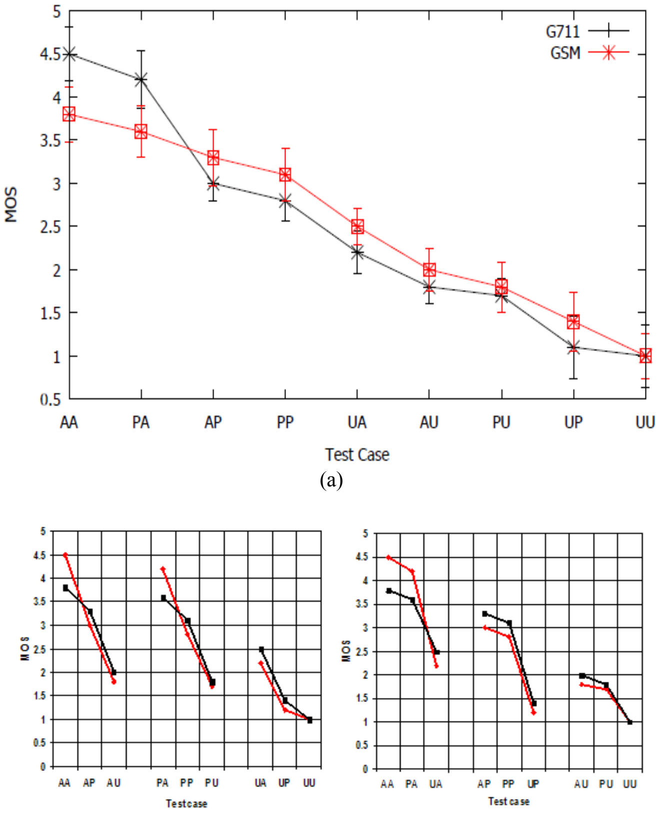
B. Closed Network Testing

A novel closed-network test methodology that involves human subjective testing is carried to understand the human perception under combination of different QoS factors. Our objective is to understand the tradeoff between the losses and delay from the human perception point of view in order to take this tradeoff into consideration when developing our redundancy control algorithm. In these tests, human subjects are asked to rank their perception QoE (MOS) of interactive VoIP calls for different ranges of packet loss rate and delay configured using Dummynet [10].

We have carried all of the test cases: <A, A>, <A, P>, <A, U>, <P, A>, <P, P>, <P, U>, <U, A>, <U, P> and <U, U>, where each test case is defined by a certain sequence of the network factor levels <$d_{network}$, $MOS_{network}$>. For example, the <A, P> test case corresponds to network conditions that results in an acceptable one way delay and Poor MOS according to the losses and codec used. We have carried out conversation-opinion subjective tests according to the procedures provided in ITU-T Recommendation P.800/P.920 [1, 11]. Our tests are made on an isolated LAN with no cross traffic. Before the test, the people participating were informed about its purpose, procedures and the benefits from the test. ITU-T recommends 16 persons as minimum number for the accuracy required for the results [11]. In order to obtain a wider range of subjective quality scores, 20 human subjects shared in the test. According to ITU Recommendation P.920 [11], our tests were based mainly on the Name-Guessing task, which is based on a question-answer game performed according to a specified protocol. A base line test with no network impairments was executed before starting the tests.

The human subjects were asked to rank their subjective perceptual quality for the test cases relative to the base line test. In our experiment, each test case is tested 5 times on different people, then we obtain the conversational MOS ranges from 1(Bad) to 5 (Excellent) as seen in Fig. 4. In each time, we have reproduced the test case by different ranges from packet loss, burst ratio and delay to lead finally to the required combination of certain <$d_{network}$, $MOS_{network}$>. The final MOS will be the arithmetic mean of all the individual scores. We noticed that our results from these tests could be generalized for any voice codec. For simplicity, we focus in our testing on 2 common used codecs: G.711 and GSM FR.

Our results are seen in Fig. 4, the test case <$d_{network}$, $MOS_{network}$> is represented on the x-axis and its corresponding average MOS is shown on the y-axis. Interestingly, our tests show that human perception is more sensitive to packet loss than delay; this can be directly observed from the higher MOS resulted from <P, A> test case than <A, P>. It was observed that the transition from acceptable level of delay to poor is not clearly evident to human perception, while the unacceptable delay level is annoying and the listeners recognized an obvious difference.



(a)

(b)

(c)

Fig. 4. Subjective MOS testing. (a) Comparison of MOS between G.711 and G723.1 codecs. (b) Packet loss transition effect. (c) Delay transition effect.

This can be seen in Fig. 4-c by the small slope in the 2 codecs between the following test cases: <A, A>&<P, A>, <A, P>& <P, P>, <A, U>&<P, U> and the higher slope between <P, A>& <U, A>, <P, P>& <U, P>, <P, U>&<U, U>. On the other hand, the transition between acceptable MOS (indicating the packet loss level) to poor then unacceptable level is easily recognized by human perception; this was shown by the large slope between <A, A> & <A, P>&<A, U>, <P, A>&<P, P> &<P, U>, <U, A> & <U, P> & <U, U> as seen Fig. 4-b. From our tests, we have noticed also that people prefer in their conversation to have both poor delay and packet loss rather than having one of them with an unacceptable level as shown in Fig. 4-a where the MOS of the <P, P> test case is greater than the <A, U>, <P, U>, <U, A> and <U, P>.

In order to evaluate the call quality based on our subjective testing, we define the term *"APU score"* as a single metric that indicates the QoE at the end user. We have given a score from 9 to 1 with single down step from <A, A> to <U, U> ranked by subjective MOS. For instance, a packet loss of 5%, burst ratio equals to 2 with G711 codec will results in a MOS of 2.48 and given the delay of 160ms; this means that this call state is <P, P> which reflects an APU score equals to 6 indicating the conversational call quality.

C. Proposed APU Algorithm

Upon our previous subjective testing results, we noticed that the tradeoff between packet losses and delay should be taken into account when developing an algorithm that will change any factor at the expense of the other. The redundancy control algorithm is an example of such an algorithm.

Based on our previous results, we deduced a state diagram that show the desirable transitions between the current state of the on-going call to a better state after applying certain FEC RS code. The APU state diagram is seen in Fig. 5. It shows only the acceptable transitions, for instance, if the on-going call is in <P, A> state indicating poor delay level and acceptable quality as a result of losses. Hence, there is no possibility to move to state <A, A> as a result of using any FEC coding scheme because the FEC will always increase the delay over-head so it is not practical scenario. Another example is that moving from <A, U> state to <A, A> is possible as the delay might increase but it can still lie in the acceptable range (e.g.: from 30 ms to 70 ms) whilst the quality as a consequence of losses could be improved but in all conditions delay couldn't be improved using FEC.

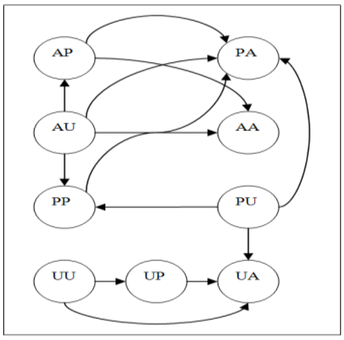


Fig. 5. APU State Diagram.

TABLE II. EXPERIMENTAL CONDITIONS

Time (sec.) Test Case		0 – 25		25 – 50		50 – 75		75 – 100		100 – 125		125 – 150		>150	
Test case	**Codec/ Burst Ratio**	Packet loss	Delay (ms)	Packet loss	Delay (ms)	Packet loss	Delay (ms)	Packet loss	Delay (ms)	Packet loss	Delay (ms)	Packet loss	Delay (ms)	Packe t loss	Delay (ms)
1st	**G.711/1.5**	0 %	3-5	15 %	170	20 %	170	15 %	330	7 %	100	20 %	330	7 %	330
2nd	**G.723.1/2.5**	0 %	3-5	10 %	330	10 %	100	10 %	170	15 %	330	15 %	1000	7 %	1000

Based on our understanding of the tradeoff between the losses, delay and their effect on the final perceived call quality; we found that switching between different RS codes might be the optimum solution for developing an adaptive redundancy control algorithm for VoIP applications. Our algorithm is based on choosing the optimum FEC coding scheme during the call from three different RS codes. We use the RTCP receiver reports (RTCP RRs) to measure different QoS parameters affecting call quality. We use one of the unused fields in RTCP RRs to include random and burst losses. In order to derive the <$d_{network}$, $MOS_{network}$> pair, we directly use the delay from the RTCP RR whilst we map the burst ratio, random packet loss and with the known codec used into MOS using the corrected simplified E-model [12]. In order to measure the expected call quality from using different RS codes, we added *nT* as a worst case delay overhead depending on the used RS code. For instance, when using RS(3, 2), n will be 3 and if G.711 codec is used with packet interval of 20ms, then T will be 20, thus we predict an overhead delay of 60ms to be added to the call session after reconstruction, whilst we predict the loss rate after reconstruction as derived in [13].

We finally compare all the expected call quality resulting from the 3 different RS codes used by our algorithm based on the APU metric we derived before. Hence, we decide the optimum code to switch to during the ongoing call. A summary of the proposed APU algorithm is shown in Fig 6.

```
if (! 5 RTCP RRs are received)
    Wait until all the 5RRs received;
else
    Calculate avgPacketLoss, avgBurstRatio, and avgDelay;
    // Calculate current state without using any RS codes:
    Calculate rating MOS using E-model ($MOS_{network}$);
    Calculate overall Delay ($d_{network}$);
    if (MOS != P||U)
        Nothing to be done;
    else
        Deduce current state <$d_{network}$, $MOS_{network}$>;
        Deduce equivalent APU score for the current state;
        Add "NO RS" and its APU score in list (validStates);
        Loop for i=0....2
            Calculate MOS after reconstruction for RS(2+i,1+i);
            Calculate overall delay after reconstruction
            Deduce next_state <$d_{network}$, $MOS_{network}$>;
            if (next_state [i] is valid transition state) // from state diagram
                Deduce APU_score for next_state[i];
                Add RS(2+i,1+i) and its APU score in list (validStates);
            else
                Exclude next_state;
            end if
        end Loop
      Sort the validStates list by APU score in descending order;
      if (2 states or more has same APU score)
          Sort them with minimum delay mode // RS(2,1)<RS(3,2)<RS(4,3)
      end if
Use the top RS mode in the validStates list;
end if
```

Fig. 6. APU Algorithm applied at the sender side.

V. SIMULATION RESULTS

In this section, we present our simulation results for the APU algorithm. We show its response under different network conditions and its effect on the overall call quality compared to the use of fixed RS codes. Our results are based on changing the network conditions 6 times with different levels of delay and packet loss. We have tested the APU algorithm under 2 commonly used codecs: G.711 and G.723.1 6.4k. The network conditions are changed just before 25, 50, 75, 100, 125 and 150 seconds from the start of the call (see Table II). We assume it takes 2 more seconds to switch between different states having different APU scores. In our results, the time is represented on the x-axis in seconds, while the APU score indicating the conversational call quality is represented on the y-axis.

A sample of our results is described for 2 different test cases, the first test case uses G.711 while the second uses G.723.1 6.4k. In each test case, we compare the performance of our adaptive APU algorithm developed with the fixed RS (2, 1), RS (3, 2) and RS (4, 3). Our experimental conditions are shown in Table II.

A. First test case

In our first test case, we used G.711 codec with burst ratio 1.5 under different percentages of packet loss and different delay levels. The response of the APU algorithm compared to the use of different fixed RS codes is shown in Fig. 7 and the analysis during the call is shown in Table III.

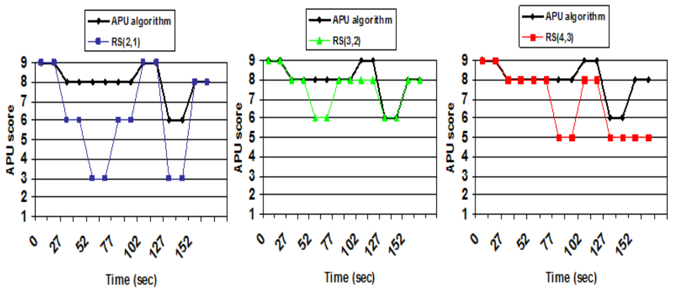


Fig. 7. Comparative Analysis of the first test case.

TABLE III. FIRST TEST CASE ANALYSIS

Time (sec.)	APU algorithm		RS(2,1)	RS(3,2)	RS(4,3)
	state	code			
0-25	AA	-	AA	AA	AA
25-50	PA	3,2	PP	PA	PA
50-75	PA	4,3	PU	PP	PA
75-100	PA	3,2	PP	PA	UA
100-125	AA	2,1	AA	PA	PA
125-150	PP	3,2	PU	PP	UA
>150	PA	2,1	PA	PA	UA

B. *Second test case*

In our second test case, we used G.723.1 6.4k codec with burst ratio 2.5 under different percentages of packet loss and different delay levels. The response of the APU algorithm compared to different pure RS codes is shown in Fig. 8 and the analysis during the call is shown in Table IV.

Our results show that our derived APU algorithm gives higher APU score indicating higher conversational call quality compared when using fixed RS codes. This can be seen from the results of the two test cases are shown in Fig. 7-Fig. 8.

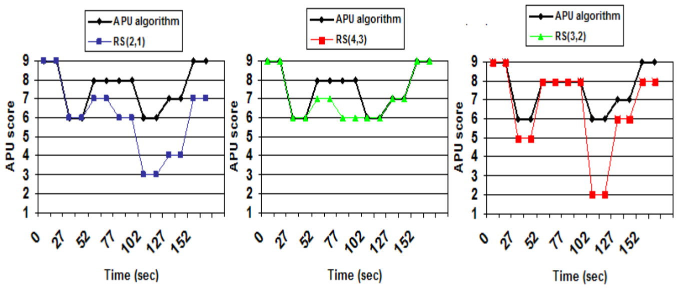


Fig. 8. Comparative Analysis of the second test case.

TABLE IV. SECOND TEST CASE ANALYSIS

Time (sec.)	APU algorithm		RS(2,1)	RS(3,2)	RS(4,3)
	state	code			
0-25	AA	-	AA	AA	AA
25-50	PP	2,1	PP	PP	UA
50-75	PA	4,3	AP	AP	PA
75-100	PA	4,3	PP	PA	PA
100-125	PP	3,2	PU	PP	UP
125-150	AP	3,2	AU	AP	PP
>150	AA	3,2	AP	AA	PA

VI. CONCLUSION

FEC is used in VoIP to decrease the effect of packet loss. Moreover, FEC increases the delay over-head. We have observed that the audio quality will not be optimal if it adjusts the redundancy ratio independent of such delay over-head that might degrade the overall conversational call quality. To solve this problem, we proposed a new adaptive FEC algorithm called the "APU algorithm". In order to derive our algorithm, we have performed subjective testing to understand the tradeoff between losses and delay at the end user. Based, on our understanding from this testing, we concluded that the delay overhead should be taken into account when adjusting the redundancy ratio. We derived a single metric called the "APU Score"; a metric for rating the conversational call quality, which takes into account this tradeoff between losses and delay. We suggested an algorithm that switches between different RS codes in order to switch between different packet loss recovery and overhead delay levels in order to attain higher overall call quality. The proposed "APU algorithm" takes into account the codec used, random packet loss, network loss burstiness and delay overhead. Finally, we show that our algorithm outperforms fixed RS codes under highly variable network conditions.

REFERENCES

[1] ITU-T, "Methods for Subjective Determination of Transmission Quality," Recommendation P.800, 1996.

[2] ——, "PESQ an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," Tech. Rep. Recommendation P.862, 2001.

[3] ——, "The E-model: a computational model for use in transmission planning," Recommendation G.107, 2009.

[4] S. Agrawal, J. Ramamirtham, and R. Rastogi, "Design of active and passive probes for VoIP service quality monitoring," in Proc. 12th International Telecommnications Network Strategy and Planning Symposium (NETWORKS 2006). IEEE, 2006, pp. 1–6.

[5] W-Jiang and H.Schulzrinne, "Comparison and Optimization of packet loss repair methodson VoIP perceived Quality under bursty loss", NOSSDAV, 2002.

[6] T-Y.Huang, P-Huang, K-T.Chen and P-J.Wang,"Could Skype be more Satisfying?", Journal IEEE Network, The Magazine of Global Internetworking, Volume 24 Issue 2 , March/April 2012, pp.42-48.

[7] W-Jiang and H.Schulzrinne, "Comparisons of EC and codec robustness on VoIP quality and bandwidth efficiency", world scientific, june 5, 2002.

[8] M.Al-Rousan, A.Nawasrah, "Adaptive FEC technique for multimedia applications over the internet", Journal of emerging technologies in web intelligence, VOL. 4, No2, May 2012.

[9] ITU-T, "Delay variation on unshared access lines", Recommendation G.114, 2009.

[10] Marta Carbona and Luigi Rizzo, "Dummynet Revisited", ACM SIGCOMM Computer Communication ReviewVolume 40 Issue 2, April 2010, 12-20.

[11] ITU-T, "Interactive test methods for audiovisual communications 2000," Recommendation P.920, 2000.

[12] H.Assem, D.Malone, J.Dunne, P.O'Sullivan, "Monitoring the VoIP call quality using Improved Simplified E-model", ICNC 2013, IEEE, Janurary, To appear.

[13] J-C. Bolot, S.Forsee-Parisis, D. Towsley, "Adaptive FEC-based error control for Internet telephony", Infocom 99, IEEE, March, pp/ 1453-1460.