From ITU-T G.722.1 to ITU-T G.722.1 Annex C: A New Low-Complexity 14kHz Bandwidth Audio Coding Standard

Minjie Xie, Dave Lindbergh*, and Peter Chu
Polycom, Inc., 10 Mall Road, Suite 110, Burlington, MA 01803, USA
Email: {Minjie.Xie, Peter.Chu}@polycom.com, lindbergh@92F1.com

Abstract— **This paper describes the low-complexity 14kHz bandwidth audio coding algorithm which has been recently standardized by ITU-T as Recommendation G.722.1 Annex C ("G.722.1C"). The algorithm is an extension to ITU-T Recommendation G.722.1 and a doubled form of the G.722.1 algorithm to permit 14 kHz audio bandwidth using a 32 kHz audio sample rate, at 24, 32, and 48 kbit/s. The G.722.1C codec features very high audio quality, extremely low computational complexity, and low algorithmic delay compared to other state-of-the-art audio coding algorithms. This codec is suitable for use in video conferencing and teleconferencing, and Internet streaming applications as well as a general-purpose 14 kHz audio codec. Subjective test results from the Characterization phase of G.722.1C are also presented in the paper.**

Index Terms— **Audio coding, low complexity, super-wideband, transform coding**

I. INTRODUCTION

As the video conferencing and teleconferencing markets mature, these applications are experiencing strong market and user pressure toward support of super-wideband audio quality.

In video conferencing applications, there is already a strong market trend toward 14 kHz audio as an improvement on 7 kHz wideband audio provided with G.722 [1] and G.722.1 [2]. Despite the relatively small amount of energy above 7 kHz in human speech, the additional octave of bandwidth represents a very significant improvement in the subjective quality and "naturalness" of speech applications (such as video conferencing), and of course is very important for music reproduction. Low complexity is especially important in video conferencing applications, as the encoding of video is extremely compute-intensive. This is becoming more challenging despite faster processors, as video resolution expectations are rapidly rising from CIF (*Common Intermediate Format*, 100k pixels) to full SD (*Standard Definition*, typically 400k pixels), to HD (*High Definition*, up to 2M pixels). At the same time, the average price of video systems has been dropping dramatically, putting more pressure on system cost. A simple audio coding algorithm can free cycles for video coding and other audio signal processing.

Teleconferencing/speakerphone applications share most of the audio characteristics of video conferences. Since the price of speakerphones, which lack the relatively expensive cameras and displays of video systems, is very sensitive to the cost of the DSP chip, the market success of super-wideband speakerphones will greatly benefit from a low-complexity coding algorithm. In this market, 7 kHz wideband capability is increasingly commonly offered, and the same market pressures that led to super-wideband (usually 14 kHz) audio for video conferencing are expected to drive speakerphones in the same direction soon. Some existing systems (mostly for IP on LANs) send uncompressed samples to avoid DSP cost, while others use proprietary compression algorithms, often of such complexity that they can be sold only at the very highest end of the market.

The urgency of the market need for such an audio coding standard was illustrated by products already in the market in 2004, from an increasing number of vendors, which offered super-wideband audio using non-interoperable proprietary schemes. Without a standardized low-complexity codec, full interoperability with low-cost devices can be difficult (requiring transcoding) or impossible.

At the June 2004 meeting of ITU-T Q.10/SG16, Polycom, Inc. proposed that ITU-T should standardize a low-complexity super-wideband audio codec [3] and submitted its Siren14™ codec, a doubled form of the algorithm of G.722.1, to ITU-T as a candidate [4]. The Siren14™ codec met all the requirements in the subjective tests for the Qualification phase in November 2004 [5] and for the Characterization phase in March 2005 [6][7], respectively. At the April 2005 meeting of ITU-T WP3/SG16, Siren14™ was adopted as ITU-T Recommendation G.722.1 Annex C [8] and became the first super-wideband audio coding standard of ITU-T.

This paper is based in part on "ITU-T G.722.1 Annex C: A New Low-complexity 14 kHz Audio Coding Standard", by Minjie Xie, Dave Lindbergh, and Peter Chu which appeared in the Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France, May 2006. © 2006 IEEE.
* Dave Lindbergh was with Polycom, Inc., when this work was done. He is now with Hooke Laboratories, Inc., Woburn, MA, 01801, USA.

In this paper, we first review the main body of ITU-T Recommendation G.722.1 and then describe the algorithm of the 14 kHz Extension Mode to ITU-T G.722.1. Finally, subjective test results from the Characterization phase of G.722.1C are summarized.

II. Review of G.722.1 Algorithm

The main body of ITU-T Recommendation G.722.1 was approved by ITU-T as an international standard in 1998 and describes a wideband coding algorithm that provides an audio bandwidth of 50 Hz to 7 kHz, operating at a bit rate of 24 kbit/s or 32 kbit/s [2]. In this section, we will review the algorithm of the main body of ITU-T G.722.1.

The codec is based on transform coding [9], using a Modulated Lapped Transform (MLT) [10] and operates on frames of 20 ms corresponding to 320 samples at a 16 kHz sampling rate. Because the transform window length is 640 samples and a 50 per cent overlap is used between frames, the effective look-ahead buffer size is 320 samples. Hence the total algorithmic delay of the coder is 40 ms.

A. Encoder

Fig. 1 shows a block diagram of the encoder. We describe the encoding principle as follows.

Modulated Lapped Transform (MLT): The MLT performs a frequency spectrum analysis on audio samples and converts the samples from the time domain into a frequency domain representation. Every 20 ms the most recent 640 audio samples are fed to the MLT and are transformed into a frame of 320 transform coefficients centered at 25 Hz intervals. The MLT is given by

$$mlt(m) = \sum_{n=0}^{639} \sqrt{\frac{2}{320}} \sin\left(\frac{\pi}{640}(n+0.5)\right) \cos\left(\frac{\pi}{320}(n-159.5)(m+0.5)\right) x(n),$$
$$0 \le m < 320 \qquad (1)$$

where $x(n)$ are audio samples and $mlt(m)$ the MLT transform coefficients.

The MLT can be decomposed into a *window-overlap-add* operation followed by a type IV Discrete Cosine Transform (DCT). The *window-overlap-add* operation is performed as follows:

$$v(n) = w(159 - n)x(159 - n) + w(160 + n)x(160 + n),$$
$$0 \le n < 160 \qquad (2)$$

$$v(n + 160) = w(319 - n)x(320 + n) - w(n)x(639 - n),$$
$$0 \le n < 160 \qquad (3)$$

where

$$w(n) = \sin\left(\frac{\pi}{640}(n+0.5)\right), \quad 0 \le n < 320. \qquad (4)$$

Combining $v(n)$ with a type IV DCT, the resulting expression for the MLT is given by

$$mlt(m) = \sum_{n=0}^{319} \sqrt{\frac{2}{320}} \cos\left(\frac{\pi}{320}(n+0.5)(m+0.5)\right) v(n),$$
$$0 \le m < 320. \qquad (5)$$

Region Power Quantization and Coding: The MLT transform coefficients are divided into 16 regions, each having 20 transform coefficients. Thus, each region represents a bandwidth of 500 Hz. As the bandwidth is 7 kHz, only the 14 lowest regions, r_0-r_{13}, are used. The 40 MLT transform coefficients representing frequencies above 7 kHz are ignored.

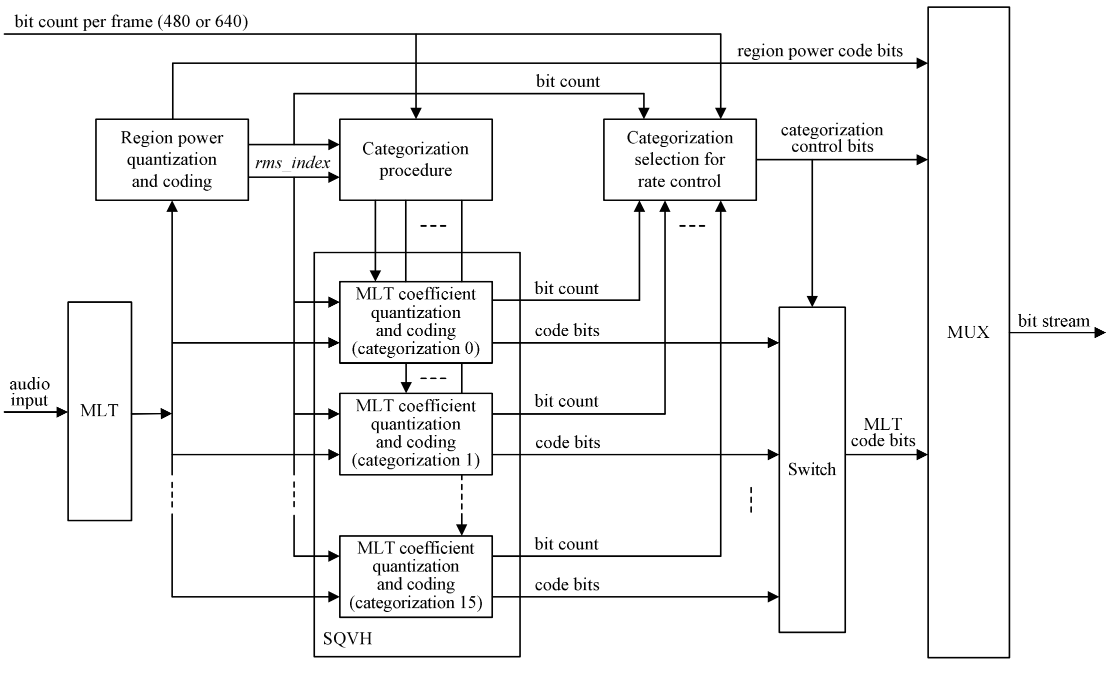

Figure 1. Block diagram of the G.722.1 encoder.

The region power or spectrum energy is determined for each of the 14 regions. The region power is defined as the root-mean-square (*rms*) value of the MLT transform coefficients in the region and is computed as

$$rms(r) = \sqrt{\frac{1}{20}\sum_{n=0}^{19} mlt(20r+n)mlt(20r+n)}, 0 \le r < 14. \quad (6)$$

The region power is then quantized with a logarithmic quantizer. The possible quantization values are the set $2^{(i/2+1)}$, where i is an integer in the range [-8, 31]. The quantization index for the lowest frequency region, *rms_index*(*0*), is further constrained to the range [1, 31].

The region power in the lowest frequency region, *rms*(*0*), is quantized with 5 bits and the quantization index *rms_index*(*0*) is directly transmitted. The quantization indices of the remaining 13 regions are differentially coded against the last highest-numbered region as

$$differential\_rms\_index(r) = rms\_index(r) - rms\_index(r-1), \quad 1 \le r < 14 \quad (7)$$

and then Huffman coded with a variable number of bits. The region power code bits are transmitted in order from the lowest frequency region to the highest. The region power quantization and coding module shown in Fig. 1 also produces the corresponding bit count for the categorization procedure described below.

Categorization: A categorization consists of one assignment for each of the 14 regions. In other words, for each of 14 regions, a categorization assigns one of the 8 categories. Using the quantized region power indices and the number of bits remaining in the frame, the categorization procedure generates 16 possible categorizations to determine the parameters used to quantize and code the MLT transform coefficients. Each categorization consists of a set of 14 "category" assignments, one assignment for each of the 14 regions. The category assigned to a region defines the quantization and coding parameters such as quantization step size, dead zone, vector dimension, and variable bit-length code for that region and also an expected total number of bits for representing the quantized MLT transform coefficients in the region. There are 8 categories: category 0 through category 7. Category 0 has the smallest quantization step size and uses the most bits. Category 7 has only one quantization output value, set to "0". The expected number of bits for each category is statistically determined and given in TABLE I.

TABLE I.
EXPECTED NUMBER OF BITS FOR EACH CATEGORY

Category	Expected number of bits per region
0	52
1	47
2	43
3	37
4	29
5	22
6	16
7	0

In the categorization procedure, only one of the 16 possible categorizations is selected to quantize the MLT transform coefficients. For each frame 4 categorization control bits are used to indicate the selected categorization and are transmitted to the decoder as side information.

After encoding the region powers, the number of available bits used for quantization of MLT coefficients, N_b, is obtained as follows:

$$N_b = \text{total bits per frame} - \text{region power code bits} - \text{categorization control bits}. \quad (8)$$

If N_b is larger than 320, the estimated number of available bits, N_{est}, is adjusted as

$$N_{est} = 320 + (N_b - 320) * 5 / 8. \quad (9)$$

With this adjustment, N_{est} is always less than N_b so to provide head room in the categorization process.

The categorization procedure begins by computing an initial categorization with an integer *offset*. For any *offset* in the range of -32 to 31, the assignment of categories is given by

$$category(r) = \text{MAX}\{0, \text{MIN}\{7, (offset - rms\_index(r))/2\}\}, \quad 0 \le r < 14 \quad (10)$$

where $0 \le category(r) < 8$,
r is the region index,
MAX(a, b) returns the largest value between a and b, and
MIN(a, b) returns the smallest value between a and b.

The expected total number of MLT code bits, N_{code}, is computed as follows:

$$N_{code} = \sum_{r=0}^{13} expected\_bits\_table(category(r)) \quad (11)$$

where *expected_bits_table*(*i*), *i*=0, 1, …, 7, is the expected number of bits for each category given in TABLE I.

The *offset* value is then adjusted until the largest *offset* found satisfies the following

$$N_{code} \ge N_{est} - 32. \quad (12)$$

Once the initial categorization is found, 15 other categorizations are derived by adjusting the category in one region per categorization. More precisely, for each new categorization the category is adjusted in only one region relative to the previous categorization. The remaining categorizations are computed as follows:

1) Compute initial category assignments, *init_categ(r),* for each region according to

$$init\_categ(r) = \text{MAX}\{0, \text{MIN}\{7, (offset - rms\_index(r))/2\}\}, \quad 0 \le r < 14 \quad (13)$$

where r is the region index,
MAX(a, b) returns the largest value between a and b, and

MIN(a, b) returns the smallest value between a and b.

2) For each region r initialize the maximum category, *max_category(r),* and the minimum category, *min_category(r)*, to *init_categ(r)*, respectively.
3) Initialize the maximum bit count, *max_bits*, and the minimum bit count, *min_bits*, to N_{code}, respectively.
4) For each of the remaining 15 categorizations, compute the sum of the maximum bit count and minimum bit count. If the sum is not larger than 2 times N_{est}, a new categorization is required with a larger expected number of bits. Otherwise, a categorization with a smaller expected number of bits is needed.
5) Compute

$$s(r) = offset - rms\_index(r) - 2 * max\_category(r), \quad 0 \le r < 14. \quad (14)$$

6) If a categorization with a larger expected number of bits is required, the lowest frequency region for which the category is not already 0 and *s(r)* is at least as small as for any other region is found. The category for this region is reduced by one in *max_category(r)*. Then the expected number of bits for this new categorization is re-computed and *max_bits* is set equal to it.
If a categorization with a smaller expected number of bits is needed, the highest frequency region for which the category is not already 7 and *s*(*r*) is at least as large as for any other region is found. The category for this region is increased by one in *min_category*(*r*). Then the expected number of bits for this new categorization is re-computed and *min_bits* is set equal to it.

In this way, 16 unique categorizations are produced and can be ordered according to their expected number of bits. Categorization 0 has the largest expected number of bits and categorization 15 the smallest. Each categorization is the same as its adjacent categorization, except in one region where the category entry will differ by one.

Scalar Quantized Vector Huffman Coding: MLT transform coefficients in categories 0 through 6 are normalized, scalar quantized, combined into vectors, and Huffman coded in the Scalar Quantized Vector Huffman Coding (SQVH) module shown in Fig. 1. As regions assigned a category 7 are quantized to "0", they are not allocated any bits for transmission.

For each region assigned a category 0 though category 6, the MLT transform coefficients are first separated into sign and magnitude (absolute value) parts. Then the magnitude parts of the MLT transform coefficients are normalized by the quantized region power, *qrms*, and scalar quantized with dead zone expansion. The dead zone is defined as the region in which the MLT transform coefficients are quantized to "0" [11]. The quantization indices, *k*, are given by

$$k(i) = \text{MIN}\{\text{whole number part of } (x * |mlt(20r+i)|) + deadzone\_rounding), kmax\}, \quad 0 \le r < 14 \text{ and } 0 \le i < 20 \quad (15)$$

where MIN(a, b) returns the smallest value between a and b, $|mlt(20r+i)|$ is the absolute value of each MLT transform coefficient, $x=1/(stepsize*qrms)$, and *stepsize, deadzone_rounding, and kmax* are given in TABLE II.

TABLE II.
CONSTANTS USED BY SQVH

Category	*stepsize*	*deadzone_rounding*	*kmax*
0	$2^{-1.5}$	0.30	13
1	$2^{-1.0}$	0.33	9
2	$2^{-0.5}$	0.36	6
3	$2^{0.0}$	0.39	4
4	$2^{0.5}$	0.42	3
5	$2^{1.0}$	0.45	2
6	$2^{1.5}$	0.50	1

The resulting quantization indices are combined into vector indices. The properties of the vectors are defined in TABLE III with the following abbreviations:

- *vd*: vector dimension
- *vpr*: number of vectors per region
- *u*: $(kmax+1)^{vd}$ which is the number of distinct vectors

TABLE III.
DEFINITION OF VECTOR PROPERTIES

Category	*vd*	*vpr*	*u*
0	2	10	196
1	2	10	100
2	2	10	49
3	4	5	625
4	4	5	256
5	5	4	243
6	5	4	32

The set of scalar values, *k*, correspond to a unique vector identified by an index as follows:

$$vector\_index(n) = \sum_{j=0}^{vd-1} k(n * vd + j)(kmax+1)^{(vd-(j+1))}, \quad 0 \le n < vpr \quad (16)$$

where n represents the n^{th} vector in the region r and j is the index to the j^{th} value of k in a given vector in the region r.

The vector indices are then Huffman coded, i.e. they are coded with a variable number of bits. The most frequent vector indices require fewer bits than the less frequent vector indices. The number of bits required to represent a vector with index *vector_index*(*n*) for a given category 0 through category 6 is provided by the SQVH bit-count tables *mlt_sqvh_bitcount_category_0*[] to *mlt_sqvh_bitcount_category_6*[], which are not given in this paper due to their sizes but can be found in [2]. The corresponding code word entries are given in the SQVH code tables *mlt_svqh_code_category_0*[] to *mlt_svqh_code_category_6*[] that also can be found in [2] as the SQVH bit-count tables. These bit counts are then summed together with the count of sign bits to determine the total number of bits required to represent the quantized MLT transform coefficients of this region with this category.

For each of the 16 possible categorizations, the total number of bits actually required to represent the frame is computed. This includes the bits used to represent the quantized region powers, the 4 categorization control bits, and the bits needed for the SQVH coding of the MLT transform coefficients. From those categorizations which yield bit totals that fit within the allotment, the categorization with the lowest index is selected. If no categorization yields a bit total that fits within the allotment, the categorization that comes closest (normally categorization15) is selected. Then, code bits are transmitted until the allotment for the frame is exhausted. It may happen that the number of bits required by the encoder to represent one 20 ms frame of audio is less than the allowed number of bits per frame (480 or 640 bits, depending on the bit-rate mode). In this case the remaining unused bits at the end of the bit stream sequence are all set to "1".

The vector indices are coded in accordance with the variable bit-length codes defined in the SQVH bit-count and SQVH code tables mentioned above. The sign bits corresponding to the non-zero MLT transform coefficients of each vector immediately follow the respective vector index code. Both vector index code bits and sign bits are transmitted in spectral frequency order from lowest to highest.

Bit Stream: The bit stream is transmitted on the channel in 3 parts: region power code bits, 4 categorization control bits, and then SQVH code bits for MLT transform coefficients. While the number of bits in a frame is fixed, except for the categorization control bits parameter, all other parameters are represented with a variable number of bits. Fig. 2 illustrates the format of the bit stream and the order of the transmitted parameter fields.

Region power code bits (variable number of bits)	Categorization control bits (4 bits)	SQVH code bits for MLT coefficients (variable number of bits)

First bit transmitted — Last bit transmitted

Figure 2. Bit stream format and transmission order.

B. Decoder

Fig. 3 shows a block diagram of the decoder. The data received at the decoder is demultiplexed into region power code bits, categorization control bits, and SQVH code bits for MLT transform coefficients.

Region Power Decoding and Reconstruction: For every frame, the first 5 bits of the bit stream are decoded to obtain the quantized region power index *rms_index*(*0*). Then, for regions 1 through 13, the variable bit-length codes for differential indices *differential_rms_index*(*r*) are decoded and the quantization indices for these region powers are reconstructed as follows:

$$rms\_index(r)=rms\_index(r-1)+differential\_rms\_index(r),\quad 1 \le r < 14. \qquad (17)$$

Categorization: After decoding the region powers, the number of bits available for representing the MLT transform coefficients is calculated as (8). Using the same categorization procedure as the encoder, the set of 16 possible categorizations computed by the encoder are reconstructed. The 4 categorization control bits indicate which categorization was used to encode the MLT transform coefficients, and consequently is also used by the decoder.

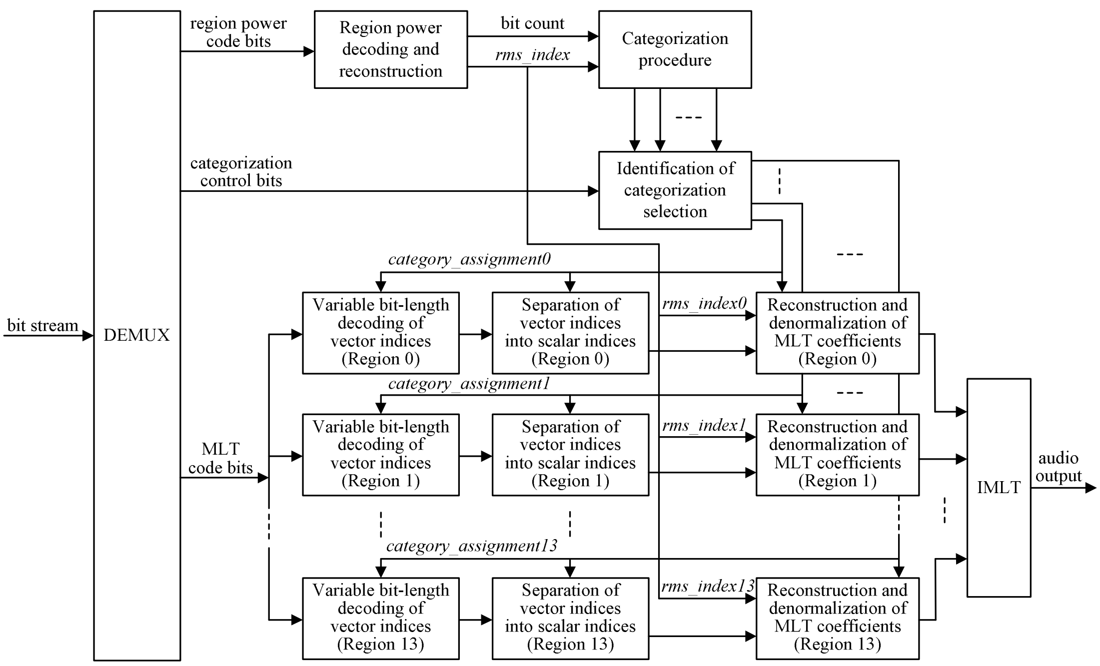

Figure 3. Block diagram of the G.722.1 decoder.

TABLE IV.
CENTROIDS USED FOR RECONSTRUCTION OF MLT TRANSFORM COEFFICIENTS

Index	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
Category0	0.000	0.392	0.761	1.120	1.477	1.832	2.183	2.541	2.893	3.245	3.598	3.942	4.288	4.724
Category1	0.000	0.544	1.060	1.563	2.068	2.571	3.072	3.562	4.070	4.620	0.000	0.000	0.000	0.000
Category2	0.000	0.746	1.464	2.180	2.882	3.584	4.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category3	0.000	1.006	2.000	2.993	3.985	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category4	0.000	1.321	2.703	3.983	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category5	0.000	1.657	3.491	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category6	0.000	1.964	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Reconstruction of MLT Coefficients: For each region, the variable bit-length codes representing the MLT vectors are decoded according to the SQVH bit-count and SQVH code tables mentioned in Section *A* for the appropriate category. The individual MLT transform coefficient quantization indices, *k*, in a region are recovered from the vector index, *vector_index*(*n*), as follows:

$$k(i) = \left\lfloor \frac{vector\_index(n)}{(kmax+1)^{j}} \right\rfloor \mathrm{mod}(kmax+1),$$

$$0 \le j < vd \text{ and } 0 \le n < vpr \qquad (18)$$

where $i = (n+1)vd - j - 1$,
$kmax$ = maximum value of k for a given category as shown in TABLE II,
$\lfloor x \rfloor$ indicates taking the greatest integer value less than or equal to x, and
mod denotes modulo operation.

The reconstruction of the MLT transform coefficients uses the centroids given in TABLE IV. The MLT coefficient amplitudes are reconstructed by computing the product of the region power in the region of interest and the centroid specified by the decoded vector index. Non-zero values have their signs set according to the sign bit. The 40 MLT transform coefficients representing frequencies above 7 kHz are set to "0".

No MLT transform coefficient amplitudes are encoded for regions assigned category 7. For categories 5 and 6 the quantization step sizes are so large that most MLT transform coefficients are coded as "0". To avoid audible artifacts, the decoder reproduces these MLT transform coefficients using "noise-fill" [12][13] for which these 0's are replaced with values of random sign and amplitude proportional to the region power for the region. The noise-fill proportionality constants are given in TABLE V.

TABLE V.
NOISE-FILL PROPORTIONALITY CONSTANTS

Category	Noise-fill proportionality constant
5	0.176777
6	0.250000
7	0.707107

Inverse Modulated Lapped Transform (IMLT): The reconstructed MLT transform coefficients are converted into time domain audio samples by an Inverse MLT (IMLT). Each IMLT operation takes in 320 MLT transform coefficients to produce 320 audio samples. The IMLT can be decomposed into a type IV DCT followed by a *window-overlap-add* operation.

The type IV DCT is defined as

$$u(n) = \sum_{m=0}^{319} \sqrt{\frac{2}{320}} \cos\left(\frac{\pi}{320}(m+0.5)(n+0.5)\right) mlt(m),$$

$$0 \le n < 320. \qquad (19)$$

The *window-overlap-add* operation uses half of the samples from the current frame's DCT output with half of those from the previous frame's DCT output, *u_old*(*n*), as follows:

$$y(n) = w(n)u(159-n) + w(319-n)u\_old(n), \quad 0 \le n < 160 \qquad (20)$$

$$y(n+160) = w(160+n)u(n) - w(159-n)u\_old(159-n), \quad 0 \le n < 160 \qquad (21)$$

where $w(n)$ is defined in (4).

For the next frame, the so far unused half of the current frame's DCT output, *u*(*n*+160), is stored in *u_old*(*n*) as follows:

$$u\_old(n) = u(n+160), \quad 0 \le n < 160. \qquad (22)$$

Frame Erasure Concealment: A frame erasure concealment procedure is incorporated into the decoder. When a frame is correctly received, the reconstructed MLT transform coefficients are stored in a buffer. If the decoder is informed that a frame has been lost or corrupted, it repeats the previous frame's reconstructed MLT transform coefficients. It proceeds by transforming them to the time domain and performing the *window-overlap-add* operation. If the previous frame was also lost or corrupted, the decoder then sets all the MLT transform coefficients in the current frame to "0".

C. Codec Complexity

TABLE VI below presents the computational complexity of G.722.1 in units of Weighted Million Operations Per Second (WMOPS). In terms of memory requirements, G.722.1 uses about 11K bytes RAM and 20K bytes ROM.

TABLE VI.
COMPUTATIONAL COMPLEXITY OF G.722.1

Bit rate (kbit/s)	Encoder (WMOPS)	Decoder (WMOPS)	Encoder+Decoder (WMOPS)
24	2.3	2.7	5.0
32	2.4	2.9	5.3

III. DESCRIPTION OF G.722.1 ANNEX C

ITU-T Recommendation G.722.1 Annex C [8] describes the 14 kHz mode at 24, 32, and 48 kbit/s for G.722.1. G.722.1C has the same algorithmic steps as the G.722.1 main body, except that the algorithm is doubled to accommodate the 14 kHz audio bandwidth.

The G.722.1C codec is designed to operate with an audio signal sampled at 32 kHz. Compared to the main body of G.722.1, this codec also operates on frames of 20 ms but the frame length is doubled from 320 samples to 640 samples due to the higher sampling frequency. The transform window size increases to 1280 samples from 640 samples in G.722.1. The algorithmic delay is still 40 ms in G.722.1C.

A. Encoder

The specific differences in the G.722.1C encoding algorithm compared to the main body of G.722.1 are as follows:

- Double the MLT transform length from 320 to 640 samples
- Double the number of frequency regions from 14 to 28
- Double the sizes of Huffman coding tables for encoding quantized region power indices
- Double the threshold for adjusting the number of available bits from 320 to 640

In G.722.1C, the MLT transforms the newest 1280 audio samples into 640 transform coefficients as follows:

$$mlt(m) = \sum_{n=0}^{1279} \sqrt{\frac{2}{640}} \sin\left(\frac{\pi}{1280}(n+0.5)\right) \cos\left(\frac{\pi}{640}(n-319.5)(m+0.5)\right) x(n),$$
$$0 \le m < 640 \qquad (23)$$

where $x(n)$ are audio samples and $mlt(m)$ the MLT transform coefficients.

The MLT can be decomposed into a *window-overlap-add* operation followed by a type IV DCT. The *window-overlap-add* operation is performed as follows:

$$v(n) = w(319-n)x(319-n) + w(320+n)x(320+n),$$
$$0 \le n < 320 \qquad (24)$$

$$v(n+320) = w(639-n)x(640+n) - w(n)x(1279-n),$$
$$0 \le n < 320 \qquad (25)$$

where

$$w(n) = \sin\left(\frac{\pi}{1280}(n+0.5)\right), \quad 0 \le n < 640 . \qquad (26)$$

Combining $v(n)$ with a type IV DCT, the resulting expression for the MLT is given by

$$mlt(m) = \sum_{n=0}^{639} \sqrt{\frac{2}{640}} \cos\left(\frac{\pi}{640}(n+0.5)(m+0.5)\right) v(n),$$
$$0 \le m < 640. \qquad (27)$$

The MLT transform coefficients are divided into 32 regions of 20 transform coefficients. Each region spans 500 Hz. Due to the 14 kHz bandwidth, only the 28 lowest regions, or 560 MLT transform coefficients, are used. The 80 MLT transform coefficients representing frequencies above 14 kHz are ignored.

For each region, the region power is computed, quantized, and coded using the same method as in the G.722.1 encoder. To encode the quantization indices of the 28 regions, the new Huffman coding tables are obtained by repeating the last row of the corresponding tables of G.722.1 fourteen times. These extended tables can be found in [8].

In this 14 kHz mode, the threshold for adjusting the number of available bits is doubled to 640. If the number of available bits, N_b, is larger than 640, the estimated number of available bits, N_{est}, is adjusted as

$$N_{est} = 640 + (N_b - 640) * 5 / 8. \qquad (28)$$

Then, the categorization procedure and SQVH coding of the MLT transform coefficients are performed in the same way as G.722.1.

For G.722.1C, the total number of bits in a frame is 480, 640, or 960 bits, corresponding to the bit rates of 24, 32, and 48 kbit/s, respectively. The bit stream is transmitted in the same format and order as G.722.1.

B. Decoder

Following are the main changes in the G.722.1C decoder when compared to G.722.1.

- Double the number of frequency regions from 14 to 28
- Double the threshold for adjusting the number of available bits from 320 to 640
- Extend the centroid table used for reconstruction of MLT transform coefficients
- Double the IMLT transform length from 320 to 640 samples

As in the G.722.1 decoder, the first 5 bits of the bit stream are assembled into the quantized region power index *rms_index(0)*. Then, the remaining 27 region powers are Huffman decoded and reconstructed.

The 4 categorization control bits are then decoded to determine which of the 16 possible categorizations was selected and transmitted by the encoder. Just as in the G.722.1 decoder, the categorization procedure uses the quantized region powers together with the number of bits remaining to be decoded in the current frame and computes the set of 16 possible categorizations.

TABLE VII.
CENTROIDS USED TO RECONSTRUCT MLT TRANSFORM COEFFICIENTS OF G.722.1C

Index	***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***
Category0	0.000	0.392	0.761	1.120	1.477	1.832	2.183	2.541	2.893	3.245	3.598	3.942	4.288	4.724	0.000	0.000
Category1	0.000	0.544	1.060	1.563	2.068	2.571	3.072	3.562	4.070	4.620	0.000	0.000	0.000	0.000	0.000	0.000
Category2	0.000	0.746	1.464	2.180	2.882	3.584	4.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category3	0.000	1.006	2.000	2.993	3.985	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category4	0.000	1.321	2.703	3.983	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category5	0.000	1.657	3.491	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Category6	0.000	1.964	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

The remaining code bits in the frame represent the quantized MLT transform coefficients and they are decoded according to the category information for each region. To reconstruct the MLT transform coefficients for the 28 regions, the centroid given in TABLE IV are extended as shown in TABLE VII. For each region, the reconstructed MLT coefficient amplitudes are obtained by taking the product of the region power with the centroid specified by the decoded vector index. The signs of non-zero amplitudes are determined by the sign bits received. The 80 MLT transform coefficients representing frequencies above 14 kHz are set to "0". The same technique of noise-fill that the G.722.1 decoder uses is applied to category 5 through category 7.

After reconstruction of the MLT transform coefficients, they are transformed into 640 time domain audio samples by an IMLT, which can be decomposed into a type IV DCT followed by a *window-overlap-add* operation.

In G.722.1C, the type IV DCT is defined as

$$u(n) = \sum_{m=0}^{639} \sqrt{\frac{2}{640}} \cos\left(\frac{\pi}{640}(m+0.5)(n+0.5)\right) mlt(m)\,,$$
$$0 \le n < 640. \qquad (29)$$

The *window-overlap-add* operation uses half of the samples from the current frame's DCT output with half of those from the previous frame's DCT output, $u\_old(n)$, as follows:

$$y(n) = w(n)u(319-n) + w(639-n)u\_old(n),$$
$$0 \le n < 320 \qquad (30)$$

$$y(n+320) = w(320+n)u(n) - w(319-n)u\_old(319-n),$$
$$0 \le n < 320 \qquad (31)$$

where $w(n)$ is defined in (26).

The unused half of the current frame's DCT output, $u(n{+}320)$, is stored as $u\_old(n)$ for the next frame:

$$u\_old(n) = u(n+320), \quad 0 \le n < 320. \qquad (32)$$

C. Codec Complexity

The computational complexity of G.722.1C in WMOPS is given in TABLE VIII. It can be seen that G.722.1C has double computational complexity in comparison with the G.722.1 main body. For memory requirements, G.722.1C needs about 18K bytes RAM and 30K bytes ROM. For purposes of comparison, TABLE IX shows the computational complexity of G.722.1C versus two well-known audio codecs, 3GPP eAAC+ [14] and 3GPP AMR-WB+ [15] in mono mode, at bit rates of 24 and 32 kbit/s and TABLE X shows the algorithmic delay of G.722.1C versus 3GPP eAAC+ and 3GPP AMR-WB+ [16]. The eAAC+ and AMR-WB+ WMOPS values were obtained by measuring the fixed-point implementations V6.0.0 on the 3GPP website as of April 2005. Sound Quality Assessment Material (SQAM) [17] speech material was used to benchmark the codecs. The 3GPP eAAC+ and AMR-WB+ encoders accept the audio files up-sampled to 48 kHz as input at 24 and 32 kbit/s. In order to encode the input signal, the encoders then convert the 48 kHz sampling frequency into an internal sampling frequency (ISF), 32 kHz for eAAC+ and 25.6 kHz for AMR-WB+.

Low complexity is a major technical advantage of G.722.1C compared to other algorithms with similar performance in this bit-rate range. It represents a separate class of audio codec for low-complexity applications such as video conferencing and teleconferencing.

TABLE VIII.
COMPUTATIONAL COMPLEXITY OF G.722.1C

Bit rate (kbit/s)	Encoder (WMOPS)	Decoder (WMOPS)	Encoder+Decoder (WMOPS)
24	4.5	5.3	9.7
32	4.8	5.5	10.3
48	5.1	5.9	10.9

TABLE IX.
COMPUTATIONAL COMPLEXITY OF G.722.1C VS. 3GPP AUDIO CODECS

Bit rate (kbit/s)	G.722.1C Enc.+Dec. (WMOPS)	eAAC+ Enc.+Dec. (WMOPS)	AMR-WB+ Enc.+Dec. (WMOPS)
24	9.7	40.8	80.1
32	10.3	42.6	86.7

TABLE X.
ALGORITHMIC DELAY OF G.722.1C VS. 3GPP AUDIO CODECS

G.722.1C (ms)	eAAC+ (ms)	AMR-WB+ (ms)
40.0	129.9[1]	113.8[2]

[1] Without bit-reservoir.
[2] ISF=25.6 kHz.

IV. Subjective Characterization Test Results

In March 2005, as a part of the G.722.1 Annex C development process in ITU-T, subjective characterization tests were performed on G.722.1C by France Telecom according to a test plan [18] designed by the ITU-T Q7/SG12 Speech Quality Experts Group (SQEG). In this section, we summarize the subjective characterization test results. More details about test design and data analysis can be found in [6][7][18][19].

The characterization test was comprised of two phases: the first for speech quality, which is related to the main anticipated applications - video conferencing and teleconferencing, and the second with music and mixed content such as film trailers, news, jingles, and advertisements.

The experiments conducted in Phase 1 were divided into two main blocks:

- Experiment 1: Single talker clean speech
- Experiment 2: Single talker with background noise including interfering talkers

Experiment 1 used the Absolute Category Rating (ACR) [20] method with the Mean Opinion Score (MOS) scale and Experiment 2 used the Degradation Category Rating (DCR) [20] method with Degradation Mean Opinion Score (DMOS).

Experiment 2 was further divided into three sub-experiments in order to test the G.722.1C codec with different types of background noise. These include "interfering talker noise" which might be experienced in a telecommunication conference in an open area and "office noise" which is typically encountered in modern office environments, as follows:

- Experiment *2a*: Reverberant speech with office noise
- Experiment *2b*: Reverberant speech with interfering talkers
- Experiment *2c*: Reverberant speech with office noise and interfering talkers

The MUlti Stimuli with Hidden Reference and Anchor points (MUSHRA) method [21] was used in Phase 2. Experiment 3 conducted in this phase consisted of one sub-experiment for each bit rate:

- Experiment *3a*: 24 kbit/s
- Experiment *3b*: 32 kbit/s
- Experiment *3c*: 48 kbit/s

All experiments were performed under error-free conditions. In each phase a floating-point version of the MPEG-4 AAC-LD [18][22] codec was used as the reference codec. The ITU-T requirement was that the G.722.1C codec in fixed-point be proven "not worse than the reference" with a 99% statistical confidence level [23]. In addition, for purposes of comparison the G.722.1C codec was also tested against the floating-point version V6.2.0 of the 3GPP eAAC+ [24] and AMR-WB+ [25] codecs at the rates of 24 and 32 kbit/s.

The subjective test results of Experiments 1 and 2 in Phase 1 [6] are summarized in TABLE XI and shown in Fig. 4-7. Subjective speech quality was evaluated by test subjects on a five-point scale, corresponding to the categories: *bad* (1), *poor* (2), *fair* (3), *good* (4), and *excellent* (5) for the MOS rating and *degradation very annoying* (1), *degradation annoying* (2), *degradation slightly annoying* (3), *degradation perceived but not annoying* (4), and *degradation not perceived or even some improvement* (5) for the DMOS rating [18]. Statistical analysis of the results [6] showed that G.722.1C met all performance requirements. In Experiments 1 and 2, G.722.1C was better than the reference codec MPEG-4 AAC-LD at 24 and 32 kbit/s and G.722.1C at 48 kbit/s was not worse than MPEG-4 AAC-LD operating at either 48 or 64 kbit/s. Compared to the other reference codecs, G.722.1C was not worse than 3GPP eAAC+ at 24 and 32 kbit/s and not worse than 3GPP AMR-WB+ in most of tests at 32 kbit/s.

The subjective test results of Experiment 3 in Phase 2 [7] are summarized in TABLE XII and shown in Fig. 8-10. Subjective quality was evaluated by test subjects on a continuous scale from 0 to 100, corresponding to the categories: *bad* (0-20), *poor* (20-40), *fair* (40-60), *good* (60-80), and *excellent* (80-100) [18]. Statistical analysis of the results [7] showed that G.722.1C met all performance requirements. In Experiment 3, G.722.1C was better than MPEG-4 AAC-LD at all bit rates and G.722.1C at 48 kbit/s was also better than MPEG-4 AAC-LD operating at 64 kbit/s.

TABLE XI.
SUBJECTIVE TEST RESULTS OF THE CHARACTERIZATION PHASE OF G.722.1C (PHASE 1)

Condition	Experiment 1 (MOS)	Experiment 2a (DMOS)	Experiment 2b (DMOS)	Experiment 2c (DMOS)
Direct	4.44	4.55	4.74	4.59
G.722.1C at 24 kbit/s	3.52	3.89	3.79	3.78
MPEG-4 AAC-LD at 24 kbit/s	2.86	2.64	2.63	2.57
3GPP AMR-WB+ at 24 kbit/s	4.11	4.17	4.36	4.17
3GPP eAAC+ at 24 kbit/s	3.67	3.48	3.81	3.76
G.722.1C at 32 kbit/s	3.80	4.18	4.24	4.16
MPEG-4 AAC-LD at 32 kbit/s	3.41	3.53	3.32	3.35
3GPP AMR-WB+ at 32 kbit/s	3.91	4.36	4.42	4.17
3GPP eAAC+ at 32 kbit/s	3.84	3.97	4.28	4.16
G.722.1C at 48 kbit/s	4.10	4.40	4.66	4.34
MPEG-4 AAC-LD at 48 kbit/s	4.12	4.37	4.50	4.32
MPEG-4 AAC-LD at 64 kbit/s	4.18	4.46	4.69	4.42

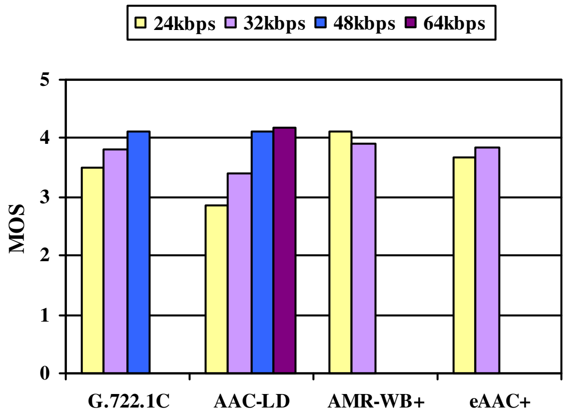

Figure 4. Clean speech.

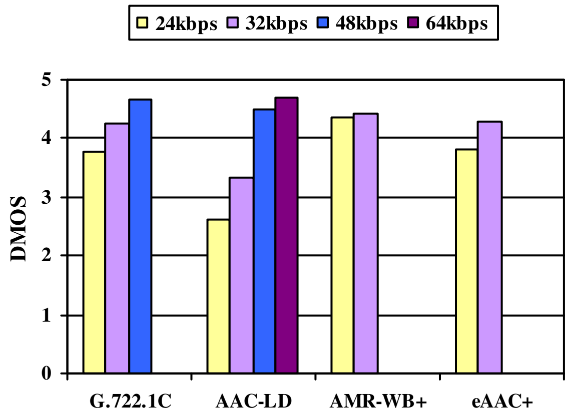

Figure 6. Reverberant speech with interfering talkers.

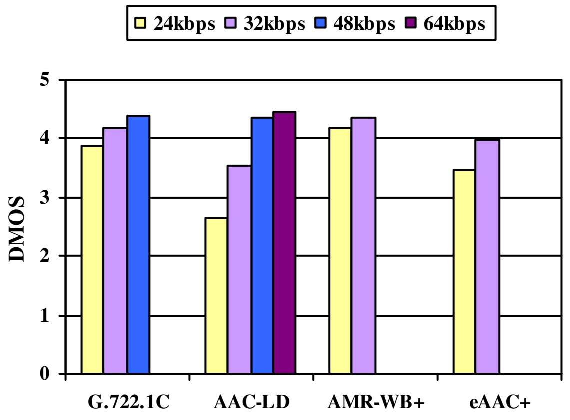

Figure 5. Reverberant speech with office noise.

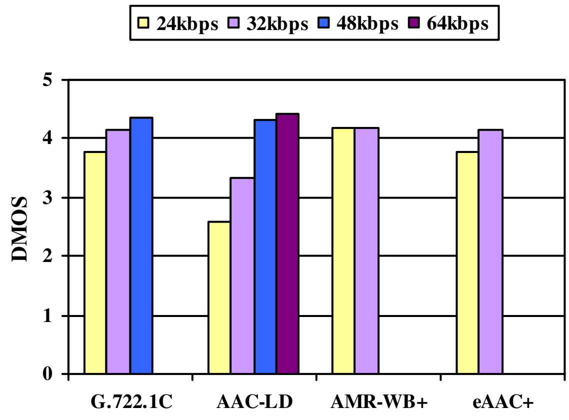

Figure 7. Reverberant speech with office noise and interfering talkers.

TABLE XII.
SUBJECTIVE TEST RESULTS OF THE CHARACTERIZATION PHASE OF G.722.1C (PHASE 2)

Condition	Experiment 3a (Mean Score)	Experiment 3b (Mean Score)	Experiment 3c (Mean Score)
G.722.1C	62.17	65.19	82.68
MPEG-4 AAC-LD	28.46	46.55	59.48
3GPP AMR-WB+	72.76	76.00	-
3GPP eAAC+	65.80	72.57	-
MPEG-4 AAC-LD at 64 kbit/s	-	-	57.84
Hidden reference (original)	98.06	96.98	97.22
10kHz anchor	71.01	59.49	63.93
7kHz anchor	35.78	31.51	33.37

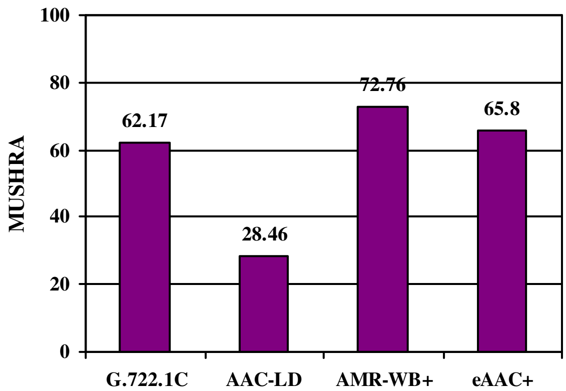

Figure 8. Music and mixed content (24 kbit/s).

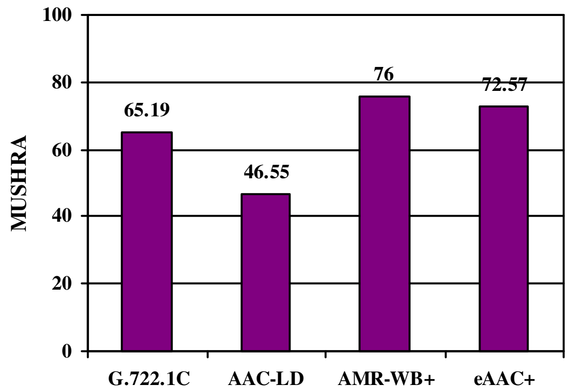

Figure 9. Music and mixed content (32 kbit/s).

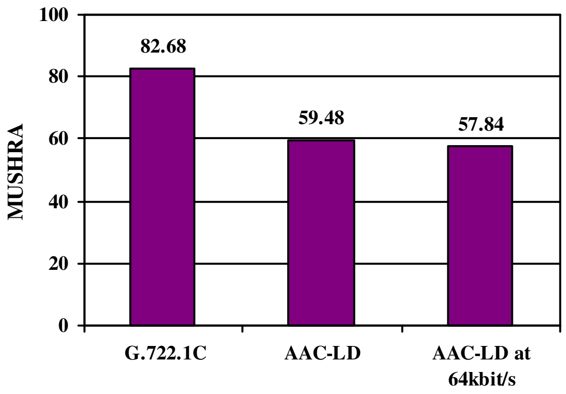

Figure 10. Music and mixed content (48 kbit/s).

V. CONCLUSION

In this paper we presented the 14 kHz audio coding algorithm and subjective characterization test results of ITU-T Recommendation G.722.1 Annex C. The G.722.1C algorithm is based on the same transform coding used in G.722.1. The differences between 7 kHz G.722.1 and the 14 kHz G.722.1C are a straightforward doubling of the algorithm. The G.722.1C codec features very high audio quality, extremely low computational complexity, and low algorithmic delay compared to other state-of-the-art audio coding codecs. Low complexity is critical for applications where other tasks such as video coding consume most of the available computing resources, or where low cost or low power consumption is important. The main intended application for this codec is videoconferencing and teleconferencing with open air microphones (including speakerphones). This codec can be also used for streaming audio over the Internet, and as a general-purpose super-wideband codec in many other applications. For purposes of evaluation, the executables and audio samples are available at http://polycom.com/Siren14/.

ACKNOWLEDGMENT

The authors would like to thank Claude Lamblin, ITU-T Q.10/SG16 Rapporteur, and Catherine Quinquis, ITU-T Q.7/SG12 Rapporteur, for their careful work guiding this project to a completion. In addition, the authors would like to thank the speech quality experts and persons who performed the subjective characterization tests at France Telecom.

REFERENCES

[1] ITU-T Rec. G.722, "7 kHz audio-coding within 64 kbit/s", November 1988.
[2] ITU-T Rec. G.722.1, "Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss", September 1999.
[3] ITU-T WP3/SG16 AC-04-Q10-06, "Proposal for Standardization of Low-Complexity Super-Wideband Audio Coding", June 2004.
[4] ITU-T WP3/SG16 AC-04-Q10-07, "Proposed new Annex C for G.722.1 – Siren14™ codec", June 2004.
[5] ITU-T WP3/SG16 TD-0021, "Qualification test results of the 14khz low-complexity audio coding algorithm at 24, 32, and 48 kbps extension to ITU-T G.722.1", November 2004.
[6] ITU-T WP3/SG16 AC-05-29, "Characterization test results of the 14 kHz low-complexity audio coding algorithm at 24, 32, and 48 kbit/s extension to ITU-T G.722.1: phase 1", April 2005.
[7] ITU-T WP3/SG16 AC-05-30, "Characterization test results of the 14 kHz low-complexity audio coding algorithm at 24, 32, and 48 kbit/s extension to ITU-T G.722.1: phase 2", April 2005.
[8] ITU-T Rec. G.722.1 Annex C, "Low complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss, Annex C: 14 kHz Mode at 24, 32, and 48 kbit/s", May 2005.
[9] N. S. Jayant and P. Noll, "*Digital Coding of Waveforms: Principles and Applications to Speech and Video*", Englewood Cliffs, NJ: Prentice-Hall, 1984.
[10] H. S. Malvar, "*Signal Processing with Lapped Transforms*", Norwood, MA: Artech House, 1992.
[11] G. J. Sullivan, "*Efficient Scalar Quantization of Exponential and Laplacian Random Variables*", IEEE Transactions on Information Theory, vol.42, No.5, pp.1365-1996, September 1996.
[12] A. Crossman, "*A Variable Bit Rate Audio Coder for Videoconferencing*", in Proceedings of the 1993 IEEE Workshop on Speech Coding for Telecommunications, pp.29-30, October 1993.
[13] A. Crossman, "*A Fixed Bit Rate Wideband Codec for Use in ISDN Videotelephony*", in Proceedings of the 1995 IEEE Workshop on Speech Coding for Telecommunications, pp.29-30, September 1995.
[14] 3GPP TS 26.411 V6.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General audio codec audio processing functions; Enhanced aacPlus general audio codec; Fixed-point ANSI-C code", March 2005.
[15] 3GPP TS 26.273 V6.0.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; ANSI-C code for the Fixed-point Extended Adaptive Multi-Rate-Wideband (AMR-WB+) codec", March 2005.
[16] 3GPP TS 26.936 V6.1.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Performance characterization of 3GPP audio codecs", March 2006.
[17] http://sound.media.mit.edu/mpeg4/audio/sqam/.
[18] ITU-T WP3/SG16 AC-05-17, "Characterization Listening Test Plan of the 14kHz Low-Complexity Audio Coding Algorithm at 24, 32 and 48 kbps Extension to ITU-T G.722.1", April 2005.
[19] ITU-T WP3/SG16 AC-05-16, "Processing Test Plan of the 14kHz Low-Complexity Audio Coding Algorithm at 24, 32 and 48 kbps Extension to ITU-T G.722.1", April 2005.
[20] ITU-T Rec. P.800, "Methods for Subjective Determination of Transmission Quality", August 1996.
[21] ITU-R Rec. BS.1534-1, "Method for the subjective assessment of intermediate quality level of coding systems", 2001-2003.
[22] ISO/IEC 14496-3:2001, "Coding of Audiovisual Objects – Part 3: Audio", 2nd Edition, 2001.
[23] ITU-T WP3/SG16 TD-0015, "Terms of Reference for the future Super Wideband Low-Complexity Audio Coding at

24, 32, and 48 kbit/s extension to ITU-T G.722.1", November 2004.
[24] 3GPP TS 26.410 V6.2.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General audio codec audio processing functions; Enhanced aacPlus general audio codec; Floating-point ANSI-C code", March 2005.
[25] 3GPP TS 26.304 V6.2.0, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Extended Adaptive Multi-Rate-Wideband (AMR-WB+) codec; Floating-point ANSI-C code", March 2005.

Minjie Xie received the B.Eng. degree in electrical engineering from Tsinghua University (Branch), Beijing, China, in 1983, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Sherbrooke, Sherbrooke (Quebec), Canada, in 1993 and 1996, respectively.

He is currently a Senior Audio Software Engineer at Polycom, Inc., Burlington, Massachusetts, USA, where he works on development of audio coding and speech processing algorithms as well as integration of real-time systems. Since 2002, he has been involved in ITU-T speech and audio coding standardization activities. His research interests include audio coding, speech processing, error concealment, vector quantization, and data compression.

Dave Lindbergh is VP Development at Hooke Laboratories, Inc., Woburn, Massachusetts, USA.

Prior to joining Hooke in 2006, he was Director of Standardization at Polycom Inc. He has been active in US and international standardization work since 1993, including Committee T1, TIA, IETF, ISO/IEC and ITU, where he was a principal contributor to ITU-T Recommendations H.223, H.224, H.281, H.460.18 and H.460.19 and V.140, served as editor for Recs. H.226, H.239, H.241, and H.324, and as chairman of the H.324 Systems Experts group, the H.264 Requirements Profiles and Levels group, and the H.264 Applications ad-hoc group. From 2002 until 2006, he was Rapporteur for ITU-T Study Group 16 Question 23, Media Coding. He is a co-author of *Digital Compression for Multimedia: Principles and Standards* (Morgan Kaufmann, 1998), a contributor to *Multimedia Communications* (Academic Press, 2001) and the *Wiley Encyclopedia of Telecommunications* (John Wiley & Sons, 2002).

Mr. Lindbergh is credited with two patents in the field of data compression and data communications. In 2002 he received an IMTC Leadership Award for his contributions to the standards community. He is an IEEE member.

Peter L. Chu received the B.S. from the Massachusetts Institute of Technology in 1975, M.S. from the University of Illinois at Urbana-Champaign in 1976, and the PhD. degree from the University of California at Berkeley in 1981, all degrees in electrical engineering. He has worked for Hewlett-Packard, Motorola-Codex, Boston University, M.I.T. Lincoln Labs, and PictureTel. Since 1999 he has been with Polycom, Inc., leading an audio research and development group. He has several patents and has published several articles in various I.E.E.E. technical journals.