



INTERNATIONAL TELECOMMUNICATION UNION

**TELECOMMUNICATION
STANDARDIZATION SECTOR**

STUDY PERIOD 1997 - 2000

COM 16-R 57-E
July 1999
Original: English

Questions: 15, 19-22/16

Texte disponible seulement en
Text available only in
Texto disponible solamente en } **E**

STUDY GROUP 16 – REPORT R 57

SOURCE*: STUDY GROUP 16 (SANTIAGO MEETING, 17-28 MAY 1999)

TITLE: PART II.B (IMPLEMENTOR'S GUIDE) OF THE REPORT OF WP 3/16 (SIGNAL PROCESSING)

CONTENTS

PART II.B - IMPLEMENTOR'S GUIDE OF RECOMMENDATIONS OF WP 3/16

	Page
1 Implementor's guide to Annexes A and B to G.723.1	2
2 Implementor's guide to Annex E to G.729	2

Attention: This is not an ITU publication made available to the public, but **an internal ITU Document** intended only for use by the Member States of the ITU and by its Sector Members and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of the ITU.

* **Contact:** TSB

Tel: +41 22 730 5860

Fax: +41 22 730 5853

email: bigi@itu.int

1 Proposed modification for G.723.1 Annexes A and B

1.1 Problem

COM 16-D302 explains a problem in the C-source code of G.723.1 Annex A, which happens during the generation of the comfort noise. A forbidden code could be generated for the LTP absolute lag (subframe 0 and 2) resulting in a memory access outside the excitation memory for the subframe 3. In the G.723.1 comfort noise excitation generation, the LTP lag is computed as $\text{lag} = \text{Olp} - 1 + \text{jitter}$, Olp being randomly generated in [123, 143] and the jitter being arbitrarily set to 1 for the subframes 0 and 2, to 0 for the subframe 1 and to 3 for the subframe 3. This implies that the lag range is [123, 143] for the subframes 0 and 2, [122, 142] for the subframe 1 and [125, 145] for the subframe 3. However, the lag must not be greater than 143 because of the memory size of the past excitation used to build the adaptive excitation. So for the subframe 3, the Olp randomly selected must not be greater than 141. The forbidden values of lag lead to a negative index in routine `Get_Rez` (in file `exc_lbc.c`) so the memory access goes backward further than allowed. However, the memory is not written but only read. Furthermore, the excitation memory is stored in the array `PrevExc` of the state variables structures `CodStat` for the encoder and `DecStat` for the decoder; so it is likely that the program reads the data in the preceding arrays of the structures that are respectively `PrevErr` in `CODSTATDEF` and `PrevLsp` in `DECSTATDEF`. However notice that this would also mean that the encoder and the decoder are slightly desynchronized.

1.2 Proposed modification

A minor correction is proposed, which is to modify a constant value in the statement selecting `Line->Olp[1]` in the routine `Calc_Exc_Rand`, i.e. to replace:

`Line->Olp[1] = random_number(21, nrandom) + (Word16)123;`

by

`Line->Olp[1] = random_number(19, nrandom) + (Word16)123;`

In the fixed-point software, this statement is at the line 86 of the file `util_cng.c` while in the floating-point software, this statement is at the line 90 of the file `utilcng2.c`. Some fixed point and floating point test sequences should also be modified.

The proposed modification is backward compatible, does not affect compatibility with the previous version, and increases interoperability.

2 Proposed modification for G.729 Annex E

COM 16-D301 explains a modification brought to the C-source code of G.729 Annex E, which only affects the frame-erasure-recovery procedure for highly stationary signals and improves the quality in case of erased frame for this type of signals.

In `decodere.c`, the variable voicing is set to zero before the call of the postfilter. In most cases, the voicing parameter will be calculated by the routine `Poste` and updated. However, if the backward mode is dominant, the postfilter is progressively reduced and at the end no postfilter is used at all and the routine `Poste` is not called so the voicing parameter takes the value zero even if the long term correlation is high. In this case, the frame erasure procedure is less efficient since it considers the signal as being unvoiced instead of voiced.

The proposed modification is the following: the voicing parameter is set to zero only if the postfilter is called, otherwise the previous value is kept. That means that in the main decoder routine (file `decodere.c`), the instruction `voicing = 0` at line 190 should be moved just after the line 224 (i.e. after the instruction : `if(ga_harm != 0)`).

The proposed modification is backward compatible, does not affect compatibility with the previous version, and increases the quality performance in case of erased frames.
