INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 1997 - 2000

COM 16-R 30-E
March 1998
Original: English

Question: 19/16

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

STUDY GROUP 16 – REPORT R 30

SOURCE*: STUDY GROUP 16 (GENEVA MEETING, 26 JANUARY - 6 FEBRUARY 1998)

TITLE: PART II.B OF THE REPORT OF WORKING PARTY 3/16 (SIGNAL PROCESSING) - IMPLEMENTOR'S GUIDE FOR ANNEX B/G.729 AND A TABLE OF G.723.1

CONTENTS

*** Contact:** TSB

Tel: +41 22 730 5860
Fax: +41 22 730 5853
E-mail: bigi@itu.int

1 Correction to Annex B of G.729

Problems related to frame erasures in G.729 Annex B

Problems related to the frame erasures simulation and bad frame handling in the case of the specific algorithms of G.729 Annex B have been discovered and are discussed below.

First the present bitstream format and frame erasure indication in the bitstream file of the ANSI C simulation software is recalled. The problem of the frame erasure indication for not transmitted frame is then addressed. Related to this problem, a bug has been found in the decoder of G.729B and G.729AB which is explained below. The test sequence TSTSEQ6.BIT designed to test the behaviour of the G.729B/G.729AB with frame erasures will also need to be modified, as explained in the last paragraph.

Bitstream format and frame erasure indication

In the distributed software of G.729 and its annexes (including G.729A, G.729B and G.729AB) the bitstream file contains, for each frame:

- one 16 bit synchro word: serial[0] = SYNC_WORD (0x6B21);
- one 16 bit word indicating the number Nb of bits for the frame: serial[1] = Nb:
  - Nb = 80 for G.729, G.729A, and for active frames of G.729B/G.729AB;
  - Nb = 15 or 16 (depending on the option of transmission) for SID frames of G.729B/G.729AB;
  - Nb = 0 for untransmitted frames of G.729B/G.729AB;
- Nb 16 bit words encoding the bits (BIT_0 = 0x007F, or BIT_1 = 0x0081)

when Nb ≠ 0, serial [i]= BIT_0 or BIT_1 for i = 2 to Nb + 1.

In the bitstream file of the G.729 software a frame erasure is indicated by zeroing those Nb words.

Frame erasure indication for not transmitted frames

This frame erasure indication does not allow to distinguish between valid not transmitted frames and erased not transmitted frames, since Nb=0 for those frames.

This is a problem since in actual implementations, both cases may occur leading possibly to different results as shown by the following example:

Suppose the following sequence of frames:

frame number	actual frame type (encoder)
T	active
(T+1)	SID
(T+2)	Not Transmitted

and suppose that frames (T+1) and (T+2) are erased.

With the frame erasure indication of the present simulation, frame (T+2) cannot be recognized as erased.

Therefore the decoder treats the frames in the following way:

frame number	present decoded frame type
T	active
(T+1)	active with BFI
(T+2)	valid not Transmitted

Yet for frame (T+2), the algorithm detects that a SID frame has been erased and creates SID parameters using the parameters stored at frame T.

But in actual system implementation, if the system is able to detect that frame (T+2) has also been erased (it is left open to system designers but some systems will make this possible), the processing would rather be:

frame number	present decoded frame type
T	active
(T+1)	active with BFI
(T+2)	active with BFI

It seems therefore necessary to have a distinct indication of the not transmitted erased frames in the bitstream file.

We propose to introduce a 4th value of the second word serial[1], RATE_0_BFI = 0x8000 for erased not transmitted frames:

Frame type	**Tx mode**	**serial[1]**
active		RATE_8000 = 80
SID	if OCTET_TX_MODE	RATE_SID_OCTET = 16
	else	RATE_SID = 15
valid not transmitted		RATE_0 = 0
erased not transmitted		RATE_0_BFI=0x8000

The number of bits of any type of frame and hence the number of words after the 2 words header into the bitstream file can then be obtained by masking the MSB of serial[1].

The erasure indication of the other types of frames could be maintained as it is (zeroing the transmitted bits) to keep the compatibility with G.729/G.729A frame erasure indication.

Error in the G.729B and G.729AB ANSI C decoder

In the G.729B/G.729 AB ANSI C simulation of the decoder, the current frame type is contained by the variable ftyp (ftyp = 0: not transmitted, ftyp = 1: active, ftyp = 2: SID) which is first filled by parm[1], deduced from the bitstream file. When a bad frame has been detected, the type of the current erased frame depends of the type of the preceding frame. This is expressed by the following lines of the source files DEC_LD8K.C/DEC_LD8A.C:

```
       if(bfi ==1)
          if(past_ftyp == 1) ftyp = 1;
          else ftyp = 0;
```

where past_ftyp contains the type of the preceding frame.

Yet the value parm[1] is not modified. In the present program, parm[1] will in fact contain the frame type of the erased frame, because parm[1] is deduced from serial[1] mentioned above, and serial[1] is not affected by the frame erasures.

The function Dec_cng( ) that generates the comfort noise filters and excitation signal is called in DEC_LD8K.C/DEC_LD8A.C for non active (SID or not transmitted) frames. But the variable ftyp is not passed as argument of this function, that uses parm[1] to identify the frame type. This is not a problem when the frame is valid or an erased not transmitted frame (in both cases parm[1]=ftyp) but creates a bug whenever the frame is an erased SID frame.

It is therefore necessary to modify the lines above in the following manner:

```
       if(bfi ==1) {
          if(past_ftyp == 1) ftyp = 1;
          else ftyp = 0;
          *parm = ftyp;
       }
```

Problem with test sequence TSTSEQ6.BIT

The test sequence TSTSEQ6.BIT is dedicated to the testing of the decoder DTX/CNG algorithms when the first SID frame of an inactive period is erased. The simulation of erased frames employed to design this test sequence was incorrect, which explains that the bug described above has not been detected. A whole SID frame (including the two words header) has been cleared instead of just the words representing the bits. Then this null frame is interpreted by the decoder as a series of 8 not transmitted frames (because in OCTET_TX_MODE). Hence the bitstream of the following frame does not loose its synchronization.

The first not transmitted frame occurs after an active frame and the procedure that creates SID parameters from previously stored values is therefore activated and tested by TESTSEQ6.BIT.

Yet this sequence is impossible from a system point of view (there should be a "BFI" somewhere between the active and the not transmitted frame). It should be corrected and this modified sequence could then also incorporate some testing of the new "not transmitted and erased" indication mentioned above.

2 Error correction in Table of G.723.1

Summary

In this contribution we describe an error in the Table of G.723.1 which specifies the octet bit packing for the high-rate (6.3 kbit/s) codec, and propose a correction to the error.

Introduction

In March 1996, SG 15 approved Recommendation G.723.1. In its Table 5, on page 23, the octet bit packing for the high-rate (6.3 kbit/s) codec is specified. Recently, it was pointed out in one of the mailing lists discussing G.723.1 packetization, that there was an error in that table. The error was initially spotted by Mr. Terry Lyons of Lucent Technologies.

Description

The problem is with octets 16-18. As things read now:

- Octet 16: POS1_B2, POS1_B0, POS0_B15..B10.
- Octet 17: POS1_B10..B3.
- Octet 18: POS2_B3..B0, POS1_B13..B11.

Hence:

- The transmitter omits POS1_B1.
- Octet 18 contains only 7 bits!

Solution

Looking at the reference C code, all the bits of the encoded bitstream are transmitted always from the least significant bit towards the most significant bit, systematically. Therefore, since the C code takes precedence over the written text, Table 5/G.723.1 should be amended such that:

- Octet 16: POS1_B1..B0, POS0_B15..B10.
- Octet 17: POS1_B9..B2.
- Octet 18: POS2_B3..B0, POS1_B13..B10.

This is done in the replacement Table below. This should be published in an Implementor's Guide at the next SG 16 meeting in January 1998.

CORRECTED TABLE 5/G.723.1

Octet bit packing for the high bit rate codec

High rate

Transmitted octets	**PAR*x* B*y*, ...**
1	LPC_B5...LPC_B0, VADFLAG_B0, RATEFLAG_B0
2	LPC_B13...LPC_B6
3	LPC_B21...LPC_B14
4	ACL0_B5...ACL0_B0, LPC_B23, LPC_B22
5	ACL2_B4...ACL2_B0, ACL1_B1, ACL1_B0, ACL0_B6
6	GAIN0_B3...GAIN0_B0, ACL3_B1, ACL3_B0, ACL2_B6, ACL2_B5
7	GAIN0_B11...GAIN0_B4
8	GAIN1_B7...GAIN1_B0
9	GAIN2_B3...GAIN2_B0, GAIN1_B11...GAIN1_B8
10	GAIN2_B11...GAIN2_B4
11	GAIN3_B7...GAIN3_B0
12	GRID3_B0, GRID2_B0, GRID1_B0, GRID0_B0, GAIN3_B11...GAIN3_B8
13	MSBPOS_B6...MSBPOS_B0, UB
14	POS0_B1. POS0_B0, MSBPOS_B12...MSBPOS_B7
15	POS0_B9...POS0_B2
16	*POS1_B1*, POS1_B0, POS0_B15...POS0_B10
17	*POS1_B9...POS1_B2*
18	POS2_B3...POS2_B0, POS1_B13...*POS1_B10*
19	POS2_B11...POS2_B4
20	POS3_B3...POS3_B0, POS2_B15...POS2_B12
21	POS3_B11...POS3_B4
22	PSIG0_B5...PSIG0_B0, POS3_B13, POS3_B12
23	PSIG2_B2...PSIG2_B0, PSIG1_B4...PSIG1_B0
24	PSIG3_B4...PSIG3_B0, PSIG2_B5...PSIG2_B3