3GPP TS 26.444 V16.5.0 (2021-12)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Codec for Enhanced Voice Services (EVS);

Test Sequences

(Release 16)

 

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP..  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

UMTS, LTE, EVS, telephony, codec

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2021, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword 4

1 Scope 5

2 References 5

3 Definitions and abbreviations 6

3.1 Definitions 6

3.2 Abbreviations 6

4 General 6

4.1 Introduction 6

5 Test sequence format 7

5.1 Introduction to test sequence format 7

5.2 File format 7

6 EVS codec test sequences including error concealment of lost packets 7

6.1 Introduction to test sequences 7

6.2 Codec configuration 7

6.3 EVS codec test sequences 7

6.3.1 EVS encoder test sequences 7

6.3.2 EVS decoder test sequences 7

6.3.3 Test sequences for AMR-WB interoperable function 8

6.3.4 Test sequences for jitter buffer management 8

7 Conformance Testing 8

7.1 Bit-exact Conformance 8

7.2 Non-Bit-exact Conformance 8

7.2.1 Overview 8

7.2.2 Running the Tests 9

Annex A: Tools Description (normative) 10

A.1 Decoder Test 10

A.1.1 General Considerations 10

A.1.2 Metrics 10

A.1.2.1 RMS Error Threshold 10

A.1.2.2 Signal to Noise Ratio (SNR) 11

A.1.2.3 Spectral Distortion 11

A.1.3 Analysis Flow and Reporting 12

A.2 Encoder Test 14

A.2.1 General Consideration 14

A.2.2 Metrics 14

A.3 MOS-LQO Test 15

A.3.1 General consideration 15

A.3.2 Test Files 17

A.3.3 Test Conditions 18

A.3.4 Thresholds and Criteria 18

A.4 Reference Implementations 19

Annex B (informative): Change history 20

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# 1 Scope

The present document specifies the digital test sequences for the Enhanced Voice Services (EVS) Codec. These sequences are used in conformance testing for implementations of the EVS Codec (3GPP TS 26.445), Voice Activity Detection (VAD) (3GPP TS.26.451), Comfort Noise Generation (3GPP TS 26.449), Discontinuous Transmission (DTX) (3GPP TS 26.450), Error Concealment of Lost Packets (3GPP TS 26.447), Jitter Buffer Management (JBM) (3GPP TS 26.448), and AMR-WB Interoperable Function (3GPP TS 26.446). In addition, the present document specifies procedures for conformance testing.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.445: "Codec for Enhanced Voice Services (EVS); Detailed Algorithmic Description".

[3] 3GPP TS 26.451: "Codec for Enhanced Voice Services (EVS); Voice Activity Detection (VAD)".

[4] 3GPP TS 26.449: "Codec for Enhanced Voice Services (EVS); Comfort Noise Generation (CNG) Aspects".

[5] 3GPP TS 26.450: "Codec for Enhanced Voice Services (EVS); Discontinuous Transmission (DTX)".

[6] 3GPP TS 26.447: "Codec for Enhanced Voice Services (EVS); Error Concealment of Lost Packets".

[7] 3GPP TS 26.442: " Codec for Enhanced Voice Services (EVS); ANSI C code (fixed-point)".

[8] 3GPP TS 26.443: "Codec for Enhanced Voice Services (EVS); ANSI C code (floating-point)".

[9] 3GPP TS 26.174: "Adaptive Multi-Rate - Wideband (AMR-WB) Speech Codec Test Sequences".

[10] 3GPP TS 26.446: "Codec for Enhanced Voice Services (EVS); AMR-WB Backward Compatible Functions".

[11] 3GPP TS 26. 448: "Codec for Enhanced Voice Services (EVS); Jitter Buffer Management".

[12] 3GPP TS 26.452: "Codec for Enhanced Voice Services (EVS); ANSI C code; Alternative fixed-point using updated basic operators".

[13] 3GPP TS 26.406: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Conformance testing".

[14] 3GPP TS 26.274: "Audio codec processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) speech codec; Conformance testing".

[15] 3GPP TS 26.952: "Codec for Enhanced Voice Services (EVS); Performance Characterization".

[16] ITU-T Recommendation P.863 (03/2018): "Perceptual objective listening quality assessment".

[17] ITU-T Recommendation P.501 (03/2017): "Test signals for use in telephonometry".

[18] ITU-R Recommendation BS.1387-1 (11/2001): "Method for objective measurements of perceived audio quality".

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 26.445 [2], 3GPP TS 26.451 [3], 3GPP TS 26.449 [4], 3GPP TS 26.450 [5], 3GPP TS 26.447 [6], 3GPP TS 26.448 [11], and 3GPP TS 26.446 [10] apply.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

ACELP Algebraic Code-Excited Linear Prediction

AMR-WB Adaptive Multi Rate Wideband (codec)

CNG Comfort Noise Generator

DTX Discontinuous Transmission

EVS Enhanced Voice Services

FB Fullband

FEC Frame Erasure Concealment

IP Internet Protocol

JBM Jitter Buffer Management

MSB Most Significant Bit

MTSI Multimedia Telephony Service for IMS

NB Narrowband

PS Packet Switched

PSTN Public Switched Telephone Network

SAD Sound Activity Detection

SC-VBR Source Controlled - Variable Bit Rate

SID Silence Insertion Descriptor

SWB Super Wideband

VAD Voice Activity Detection

WB Wideband

WMOPS Weighted Millions of Operations Per Second

# 4 General

## 4.1 Introduction

This specification provides digital test sequences which are necessary to test conformance for an implementation of the EVS codec (TS 26.445 [2]), Voice Activity Detection (TS 26.451 [3]), Comfort Noise Generation (TS 26.449 [4]), Discontinuous Transmission (TS 26.450 [5]), Concealment of Lost Packets (3GPP TS 26.447 [6]), Jitter Buffer Management (JBM) (3GPP TS 26.448 [11]) and AMR-WB Interoperable Function (3GPP TS 26.446 [10]), and for the testing of conformance for implementations of the ANSI C code in TS 26.442 [7], TS 26.443 [8] and TS 26.452 [12].

A standard compliant implementation of the above specifications shall pass conformance tests according to clause 7. The necessary test sequences can be found in the corresponding ZIP files according to the attached Readme.txt file.

Clause 5 describes the format of the files, which contain the digital test sequences. Clause 6 describes the test sequences for the EVS codec, including error concealment of lost packets, the AMR-WB interoperable function. the VAD, comfort noise generation, discontinuous transmission, the AMR-WB interoperable function, the EVS jitter buffer management, Clause 7 describes the conformance testing for implementations of the EVS codec.

# 5 Test sequence format

## 5.1 Introduction to test sequence format

This clause provides information on the format of the digital test sequences for the EVS codec (TS 26.445 [2]), Voice Activity Detection (TS 26.451 [3]), Comfort Noise Generation (TS 26.449 [4]), Discontinuous Transmission (TS 26.449 [5]), and Error Concealment of Lost Packets (TS 26.447 [6]), Jitter Buffer Management (JBM) (3GPP TS 26.448 [11]), and AMR-WB Interoperable Function (3GPP TS 26.446 [10]).

## 5.2 File format

The test sequence files in PC (little-endian) byte order are provided in archive files (ZIP format, see the pointer file Readme.txt which accompanies the present document).

Following decompression, three types of file are provided:

- Files for input to the speech encoder: \*.INP

- Files for comparison with the encoder output and for input to the speech decoder: \*.COD

- Files for comparison with the decoder output: \*.OUT

- Files for input to the speech decoder with JBM \*.RTP

- One mode control file for the mode switching test \*.MOD

- Instructions how to operate the test sequences \*TXT

# 6 EVS codec test sequences including error concealment of lost packets

## 6.1 Introduction to test sequences

This clause provides information on the test sequences designed to exercise the EVS codec (TS 26.445 [2]).

## 6.2 Codec configuration

The speech encoder shall be configured as instructed in the readme file attached.

## 6.3 EVS codec test sequences

### 6.3.1 EVS encoder test sequences

The test sequences are provided and described in the ZIP archive.

The test sequences for encoder testing and instructions to operate the encoder are summarized in Readme\_EVS\_enc.txt.

### 6.3.2 EVS decoder test sequences

The test sequences for decoder testing and instructions to operate the decoder are summarized in Readme\_EVS\_dec.txt.

### 6.3.3 Test sequences for AMR-WB interoperable function

The test sequences for the AMR-WB interoperable function include the test sequences defined in TS 26.174 [9], but \*.COD and \*.OUT files are not identical to those provided by TS 26.174.

Readme\_AMRWB\_IO\_enc.txt and Readme\_AMRWB\_IO\_dec.txt summarized the input test sequences, output test sequences, and instructions to execute the AMR-WB interoperable function test.

### 6.3.4 Test sequences for jitter buffer management

The input test sequences, the output sequences, and instructions to run the jitter buffer management test are summarized in the Readme\_JBM\_dec.txt.

# 7 Conformance Testing

## 7.1 Bit-exact Conformance

For an implementation to be declared conformant according to the bit-exact conformance method, the encoder and decoder output sequences of the implementation shall match bit-exactly the test sequences provided in the corresponding ZIP files according to the attached Readme.txt file and in accordance with clause 6. This applies for all implementations based on TS 26.442 [7], TS 26.443 [8] or TS 26.452 [12]. For fixed-point implementations based on TS 26.442 [7] and TS 26.452 [12] this is the only conformance method.

## 7.2 Non-Bit-exact Conformance

### 7.2.1 Overview

If an implementation under test is based on the reference floating–point code (TS 26.443 [8]) and the output sequences are not bit-exact to the test sequences according to clause 6, the non bit-exact conformance testing process defined here shall be used to test the conformance.

A conformant floating-point implementation of the EVS codec shall be compliant to the reference specification in TS 26.443 [8] by implementing all the algorithmic steps of the EVS codec, further specified in 3GPP TS 26.445 (Detailed Algorithmic Description) [2], 3GPP TS 26.451 (Voice Activity Detection (VAD)) [3], 3GPP TS 26.449 (Comfort Noise Generation (CNG)) [4], 3GPP TS 26.450 (Discontinuous Transmission (DTX)) [5], 3GPP TS 26.447 (Packet Loss Concealment (PLC) of Lost Packets) [6], and 3GPP TS 26.446 (AMR-WB Interoperable Function) [10].

If a floating-point implementation uses the Jitter Buffer Management (JBM) according to TS 26.448 [11], the implementation shall be compliant to the reference specification in TS 26.443 [8] by implementing all the algorithmic steps of 3GPP TS 26.448 (Jitter Buffer Management (JBM)).

An implementation shall be tested for non-bit-exact conformance using three specific tests:

- Decoder test comparing the implementation decoder with TS 26.443 [8] decoder.

- Encoder test comparing the implementation encoder with TS 26.443 [8] encoder.

- MOS-LQO verification comparing the implementation with TS 26.442 [7] implementation.

All three tests shall pass for the implementation to be declared conformant. The tests are described in more details in Annex A.

Figure 7.1 shows the flow chart of the non bit-exact conformance process.



Figure 7.1: Non bit-exact conformance process

### 7.2.2 Running the Tests

The executables and scripts illustrating how to run the tests are included in the ZIP of the floating-point test sequences. Annex A provides more details on the tests. In the case of discrepancy between the procedure described in Annex A and the scripts provided in the ZIP file, the procedure of the scripts provided prevail.

For the encoder and decoder tests, instructions on how to operate the implementation under test to run the tests are contained in the text file testvec\Readme.md. The scripts require an additional tool, ResampAudio.exe (instructions on how to download it are contained in the Readme file). The decoder and encoder test scripts are run by executing two Bourne-Shell scripts:

- Readme\_dec\_snr.sh.txt: batch file describes how to run the decoder test

- Readme\_enc\_sh.txt: batch file describes how to run the encoder test

The reference files for SNR and MLD encoder and decoder tests are also included in the zip file.

For the MOS-LQO verification, instructions on how to operate the implementation under test to run the test are contained in the text file mos-lqo\README.md. The test script is run by executing the mos-lqo.py Python® script, which in turn executes several Bourne-Shell scripts. The test script requires additional tools:

- P.863 [16] implementation, compatible with version 2 of [16]

- CopyAudio.exe (instructions on how to download it are contained in the Readme file)

For the MOS-LQO test an additional database, based on [17], [13], [14] is used, as explained in Annex A.3.2.

The implementation will be declared conformant if all three tests (encoder, decoder, MOS-LQO) are passed.

Annex A: Tools Description (normative)

# A.1 Decoder Test

## A.1.1 General Considerations

The reference PCM signals are taken from the decoded floating-point test sequences of this specification. The PCM signal under test are obtained by running the floating-point bit-stream included in this specification through the Decoder under Test (Figure A.1). The reference decoder is the floating-point code of TS 26.443 [8].



Figure A.1: Flow diagram for the decoder test using signal-based metrics

All metrics are calculated on the reference PCM signal and the PCM signal under test based on 20ms frames. The frames of the two signals will be time aligned, this means the delay compensation in EVS encoder and decoder remains ON (the default configuration). Furthermore, the frame processing is aligned with the encoded frame by adding the decoder delay. Table A.1 shows the delay values used for the different sampling frequencies.

Table A.1: Delay used for alignment of processing frames with encoded frames

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sampling frequency** | **8000 Hz** | **16000 Hz** | **32000 Hz** | **48000 Hz** |
| Delay (samples) | 10 | 37 | 74 | 111 |

The number of samples for a 20ms frame size is defined by , where represents the sampling rate.

The PCM signals and should be scaled between -1 and 1.

## A.1.2 Metrics

### A.1.2.1 RMS Error Threshold

The RMS method is derived from the decoder conformance used in ISO/IEC 14496-26 [10]. The RMS error is calculated for each 20ms frame and compared to a threshold according to:

The value chosen for the RMS error threshold is to assume change on the last bit of the audio signal:

with

### A.1.2.2 Signal to Noise Ratio (SNR)

The segmental SNR method is derived from the decoder conformance used in ISO/IEC 14496-26 [10]. For each 20 ms segment, the following values need to be calculated:

Energy of reference signal:

Noise energy:

Signal to noise ratio with

As EVS is a switched codec containing a LPC based speech coder and a MDCT based transform coder, the SNR values vary significantly depending on the used coding mode. Therefore, a constant threshold for the SNR is not suitable but instead, a reference value per frame and test vector should be specified. The SNR should be compared against the thresholds by

where is a 20 ms frame index and is the test vector index

The set of SNR reference values is included in the zip file. This set was obtained using the reference implementations listed in clause A.4.

### A.1.2.3 Spectral Distortion

The spectral distortion method can be conducted on a 20 ms frame base by the following steps:

Calculate the absolute FFT spectrum of and using a Hanning window

with

The 32768 is due to MATLAB scaling and to align to 16 bit PCM C-code. This scaling is dependent on the input value range.

For all spectral bins the distortion d is calculated according to the following pseudo code:

cnt=0  
d=0  
for k=1..N/2-1  
    if (==0 && ==0)  
        X\_Y = 1;  
        Y\_X = 1;  
    else  
        if (==0)  
            X\_Y = 0;  
            Y\_X = 2;  
        else if (==0)  
            X\_Y = 2;  
            Y\_X = 0;  
        else  
            X\_Y = ( \* ) / ( \* );  
            Y\_X = ( \* ) / ( \* );  
        end  
  end  
 COSH = (X\_Y + Y\_X - 2)/2;     
    d = d + COSH;  
    cnt = cnt+1;  
end  
d = d/cnt;

The distortion value is to be compared against a threshold . The frame will be considered as passed if

with

## A.1.3 Analysis Flow and Reporting

The three metrics are computed in a specific order, as shown in Figure A.2. Once a frame passes a metric, the process is stopped and the next frame is analysed. The SNR metric is computed on the frames failing the RMS error criteria. Similarly, the Spectral Distortion metric is computed on the frames failing the SNR criteria.



Figure A.2: Flow chart for decoder tool

In a file one or two frames could slightly be above the threshold. To avoid relaxing the threshold, a constraint on the number of frames failing per file has been added as an additional criterion.

if number\_of\_frames\_failing =< THRESH\_GOOD\_FRAMES\_TO\_PASS \* number\_of\_frame\_in file, the test signal will be considered equivalent to the reference signal.

All the test sequences need to pass for the implementation to be conformant.

In addition to the number of fail/pass test sequences, the statistics from the three methods should be displayed. Table A.2 shows an example of reporting.

Table A.2: Template for result presentation

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMS | WSNR | Spectral Distortion |
| Number of frames tested |  |  |  |
| Number of frames passing |  |  |  |
| Number of frames failing |  |  |  |
| Ratio of frames passing |  |  |  |
| Ratio of frames failing |  |  |  |

As part of conformance criteria, thresholds are set for the ratio of frames passing with RMS and WNR tests (Ratio\_RMSframespassing\_and RatioWSNRframespassing respectively).

The list of the thresholds used in decoder test are summarized in table A.3.

Table A.3: List of thresholds

|  |  |  |
| --- | --- | --- |
| Thresholds | Description | value |
| SNRHEADROOM | Headroom compare to the Tsnr threshold | 3 dB |
| CDSNRMAX | Limit of SNR for the spectral distortion test | 0 dB |
| CDSNRHEADROOM | Headroom compare to Tsnr threshold for the spectral distortion test | 10 dB |
| Tsd | Threshold for the spectral distance | 6.6 |
| THRESH\_GOOD\_FRAMES\_TO\_PASS | Factor for number of failing frame per file | 0.005 |
| Ratio\_RMSframespassing | Minimal percentage for frames passing RMS error test | 47% |
| RatioWSNRframespassing | Minimal percentage for frames passing WSNR test | 95% |

# A.2 Encoder Test

## A.2.1 General Consideration

The MLD metrics is used to test the floating-point encoder implementation. Figure A.3 shows the flow diagram of the proposed encoder conformance test:



Figure A.3: Flow diagram for the encoder test using MLD Loudness Difference metric

All encoder test sequences from this specification will be encoded using the float encoder implementation under test. The bit-stream obtained will be then decoded using the 3GPP reference float decoder from TS 26.443 [8] to obtain the test signals. The test signals will then be compared with the reference signal from this specification. Since the loudness tool (presented in clause A.2.2) operates on 48 kHz sample rate only, additional resampling is applied before processing.

## A.2.2 Metrics

The Loudness Difference (LD) is used as to assess the encoder implementation. The procedure is adopted from the loudness calculation of PEAQ [18] using the Filter bank-based ear model and follows the following steps:

- Scaling to playback level (Annex 2 section 2.2.3 of [18])

- playback level set to Lmax=103.7535 dBSPL

- Filterbank (Annex 2 section 2.2.5 of [18]):- subsample factor F changed to 16 for higher time resolution.

- Outer and Middle Ear Filtering (Annex 2 section 2.2.6 of [18])

- Frequency Domain Smearing (Annex 2 section 2.2.7 of [18])

- Rectification (Annex 2 section 2.2.8 of [18])

- Time Domain Smearing 1 - Backward Masking (Annex 2 section 2.2.9 of [18])

- Adding of Internal Noise (Annex 2 section 2.2.10 of [18])

- Time Domain Smearing 2 - Forward Masking (Annex 2 section 2.2.11 of [18])

- Loudness (Annex 2 section 3.3 of [18]):

- This section defines the specific loudness patterns for subbands and time samples

- The specific loudness patterns are calculated for:

- reference signal

- signal under test

- Loudness Difference (LD):

- The loudness difference is calculated as follows:

The LD is computed with a granularity of 2ms. To get a MLD value every 20ms, 10 segments are combined using the maximum value of those 10 segments.

Note: In the context of this test, the "loudness difference" is calculated as difference in sone values (subtraction, not division).

The LD should be below a threshold based on a reference value LDref plus some headroom.

The headroom is defined as fixed value, currently set to 1.

Then for each file a MLD could be defined as

The test file will be considered equivalent to the reference file if the MLD is positive or equal to 0, i.e. the Loudness Difference does not exceed the threshold for all the frames.

LDref has been obtained using reference implementations listed in clause A.4:

For each frame, the maximum LD value of the reference implementations defines a corridor, the ‘refline’. This then leads to a profile for each EVS test vector, which contains on a 20ms frame basis an allowed value relative to the reference. Allowed differences in implementations under test (IuTs) are thus limited to the tolerable differences by the different compilers used for the refline generation.

All the test sequences need to pass for the implementation to be conformant.

# A.3 MOS-LQO Test

## A.3.1 General consideration

For this test P.863 [16] is used. An implementation of [16] pertaining to the 2014 version 2 was used to determine the threshold values to be met.

The audio database, is based on ITU-T P.501 [17] Annex B & C items and mixed/music items as detailed in clause A.3.2. For speech with background noise pre-mixed items based on the same speech samples are used.

This test was used in EVS characterization reported in TS 26.952 [15].

For this test, four combinations of encoder/decoder are used (3GPP EVS fixed-point encoder/decoder executables are taken from TS 26.442 [7]):

a) 3GPP fixed-point encoder and 3GPP fixed-point decoder (FX/FX),

b) floating-point Encoder under Test and floating-point Decoder under Test (FL/FL),

c) 3GPP fixed-point encoder and floating-point Decoder under Test (FX/FL),

d) floating-point Encoder under Test and 3GPP fixed-point decoder (FL/FX).

The MOS-LQO scores are computed for each of the four cases using the decoded files and the original test files.

30 files representing various talkers and languages are used for each speech test condition, and the average MOS-LQO scores are reported. In addition, 30 mixed/music files are used for the non-speech test conditions as decribed in clause A.3.2.

The scenario a) is considered the reference score. For the three other scenarios (b, c and d), the difference in MOS-LQO of a) are then computed:

- a) - b)

- a) - c)

- a) - d)

The difference a) - b) assesses the encoder + decoder floating-point implementation, the difference a) - c) assesses the decoder implementation and a) - d) assesses the encoder implementation.

Figures A.4, A.5, and A.6 represent the flow diagram to obtain the MOS-LQO in the three scenarios.



Figure A.4: Flow diagram to obtain the MOS-LQO for floating-point Encoder under Test and floating-point Decoder under Test



Figure A.5: Flow diagram to obtain the MOS-LQO for 3GPP fixed-point Encoder and floating-point Decoder under Test



Figure A.6: Flow diagram to obtain the MOS-LQO for floating-point Encoder under Test and 3GPP fixed-point decoder

## A.3.2 Test Files

The test files are based on ITU-T P.501 [17] Annex B & C. 30 files representing various talkers and languages. The files are listed below:

an1f1s1 => AnnexC//P501\_C\_chinese\_f1\_FB\_48k.wav

an1f1s2 => AnnexC//P501\_C\_chinese\_f2\_FB\_48k.wav

an1f1s3 => Speech and Noise Signals Clause B/Dutch\_FB\_clause\_B.3.2//female 1.wav

an1f1s4 => Speech and Noise Signals Clause B/Dutch\_FB\_clause\_B.3.2//female 2.wav

an1f1s5 => AnnexC//P501\_C\_english\_f1\_FB\_48k.wav

an1f2s1 => AnnexC//P501\_C\_english\_f2\_FB\_48k.wav

an1f2s2 => AnnexC//P501\_C\_finnish\_f1\_FB\_48k.wav

an1f2s3 => AnnexC//P501\_C\_finnish\_f2\_FB\_48k.wav

an1f2s4 => AnnexC//P501\_C\_french\_f1\_FB\_48k.wav

an1f2s5 => AnnexC//P501\_C\_french\_f2\_FB\_48k.wav

an1f3s1 => AnnexC//P501\_C\_german\_f1\_FB\_48k.wav

an1f3s2 => AnnexC//P501\_C\_german\_f2\_FB\_48k.wav

an1f3s3 => AnnexC//P501\_C\_italian\_f1\_FB\_48k.wav

an1f3s4 => AnnexC//P501\_C\_italian\_f2\_FB\_48k.wav

an1f3s5 => AnnexC//P501\_C\_japanese\_f1\_FB\_48k.wav

an1m1s1 => AnnexC//P501\_C\_chinese\_m1\_FB\_48k.wav

an1m1s2 => AnnexC//P501\_C\_chinese\_m2\_FB\_48k.wav

an1m1s3 => Speech and Noise Signals Clause B/Dutch\_FB\_clause\_B.3.2//male 1.wav

an1m1s4 => Speech and Noise Signals Clause B/Dutch\_FB\_clause\_B.3.2//male 2.wav

an1m1s5 => AnnexC//P501\_C\_english\_m1\_FB\_48k.wav

an1m2s1 => AnnexC//P501\_C\_english\_m2\_FB\_48k.wav

an1m2s2 => AnnexC//P501\_C\_finnish\_m1\_FB\_48k.wav

an1m2s3 => AnnexC//P501\_C\_finnish\_m2\_FB\_48k.wav

an1m2s4 => AnnexC//P501\_C\_french\_m1\_FB\_48k.wav

an1m2s5 => AnnexC//P501\_C\_french\_m2\_FB\_48k.wav

an1m3s1 => AnnexC//P501\_C\_german\_m1\_FB\_48k.wav

an1m3s2 => AnnexC//P501\_C\_german\_m2\_FB\_48k.wav

an1m3s3 => AnnexC//P501\_C\_italian\_m1\_FB\_48k.wav

an1m3s4 => AnnexC//P501\_C\_italian\_m2\_FB\_48k.wav

an1m3s5 => AnnexC//P501\_C\_japanese\_m1\_FB\_48k.wav

The noisy speech items are created from the clean speech items above and mixed with car, street, or office noise.

The mixed content and music items are selected from [13], and [14] as follows:

an1a1s1 => samples 840000:1104000 from {26444}/stv48c.INP,

an1a1s2 => samples 1260000:1404288 from{26444}/stv48c.INP,

an1a1s3 => samples 1611888:1793270 from {26444}/stv48c.INP,

an1a1s4 => samples 1793270:2057040 from {26444}/stv48c.INP,

an1a1s5 => left channel from {26406}/guitar\_cymbals.wav,

an1a2s1 => left channel from {26274}/m\_cl\_x\_1\_org.wav,

an1a2s2 => left channel from {26274}/m\_cl\_x\_2\_org.wav,

an1a2s3 => left channel from {26274}/m\_ot\_x\_5\_org.wav,

an1a2s4 => left channel from {26274}/m\_ot\_x\_8\_org.wav,

an1a2s5 => left channel from {26274}/m\_si\_x\_3\_org.wav,

an1a3s1 => left channel from {26274}/m\_ch\_x\_1\_org.wav,

an1a3s2 => left channel from {26274}/m\_po\_x\_2\_org.wav,

an1a3s3 => left channel from {26274}/m\_ot\_x\_4\_org.wav,

an1a3s4 => left channel from {26274}/m\_ot\_x\_9\_org.wav,

an1a3s5 => left channel from {26274}/m\_po\_x\_3\_org.wav,

an1a4s1 => left channel from {26274}/m\_ot\_x\_6\_org.wav,

an1a4s2 => left channel from {26274}/m\_ot\_x\_3\_org.wav,

an1a4s3 => left channel from {26274}/m\_ot\_x\_a\_org.wav,

an1a4s4 => left channel from {26274}/m\_ot\_x\_b.org.wav,

an1a4s5 => left channel from {26406}/hihat.wav,

an1a5s1 => left channel from {26274}/m\_ot\_x\_7\_org.wav,

an1a5s2 => left channel from {26274}/m\_po\_x\_1\_org.wav,

an1a5s3 => left channel from {26274}/m\_ot\_x\_2\_org.wav,

an1a5s4 => left channel from {26274}/m\_po\_x\_4\_org.wav,

an1a5s5 => left channel from {26274}/m\_po\_x\_5\_org.wav,

an1a6s1 => left channel from {26274}/m\_po\_x\_6\_org.wav,

an1a6s2 => left channel from {26274}/m\_po\_x\_3\_org.wav,

an1a6s3 => left channel from {26274}/m\_si\_x\_1\_org.wav,

an1a6s4 => left channel from {26274}/m\_si\_x\_2\_org.wav,

an1a6s5 => left channel from {26274}/m\_vo\_x\_1\_org.wav

## A.3.3 Test Conditions

The differences are computed for various test conditions:

- All the codec modes of EVS

- All the bandwidths of EVS

- All the bit-rates of EVS, including bit-rate switching

- DTX ON and OFF

- Various levels: -26 dB, -36 dB, -16 dB

- Various noise conditions

- Various impairment conditions

The files have been processed according to EVS-7c (EVS processing plan) for the various test conditions [6]. In all, 941 test conditions are assessed.

The processing generates for all 941 test conditions from the items detailed in clause A.3.2, roughly 225000 seconds (or ~62 hours) of PCM data, which shall be assessed with P.863 [16] according to version 2 to generate the average MOS-LQO differences per test condition.

NOTE: Implementers are advised to ensure that sufficient free storage space is available as the processing may require up to 100 GB of storage. Processing and P.863 [16] evaluation may also require significant amounts of time.

## A.3.4 Thresholds and Criteria

From the MOS-LQO differences of the test condition, the average, 95%, 99% and Maximum are computed for all bandwidths combined, as well as for each set of bandwidth condition. The number of test condition for each bandwidth and the total are summarized in Table A.6.

Table A.6: Number of test conditions per bandwidth excluding the EVS JBM [11]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bandwidth | NB | WB | WBIO | SWB | FB | All |
| Number | 118 | 214 | 216 | 170 | 143 | 861 |

An implementation according to 26.443 Version 16.3.0 will be considered passing the MOS-LQO verification if all the average, 95 percentile, 99 percentile and maximum MOS-LQO differences are below the thresholds proposed in Table A.7 for all bandwidths.

Table A.7: Thresholds for MOS\_LQO difference excluding the EVS JBM [11]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| All | Average | 95% | 99% | Max |
| A-B | 0.001 | 0.04 | 0.07 | 0.11 |
| A-C | 0.001 | 0.02 | 0.04 | 0.09 |
| A-D | 0.002 | 0.04 | 0.07 | 0.12 |
| NB | Average | 95% | 99% | Max |
| A-B | 0.01 | 0.07 | 0.09 | 0.1 |
| A-C | 0.002 | 0.02 | 0.02 | 0.04 |
| A-D | 0.012 | 0.07 | 0.09 | 0.11 |
| WB | Average | 95% | 99% | Max |
| A-B | 0.002 | 0.05 | 0.06 | 0.08 |
| A-C | 0.001 | 0.02 | 0.04 | 0.09 |
| A-D | 0.004 | 0.05 | 0.07 | 0.09 |
| WBIO | Average | 95% | 99% | Max |
| A-B | 0.006 | 0.02 | 0.04 | 0.08 |
| A-C | 0.003 | 0.01 | 0.02 | 0.04 |
| A-D | 0.004 | 0.01 | 0.03 | 0.08 |
| SWB | Average | 95% | 99% | Max |
| A-B | 0.002 | 0.04 | 0.06 | 0.08 |
| A-C | 0.003 | 0.03 | 0.04 | 0.05 |
| A-D | 0.003 | 0.04 | 0.07 | 0.08 |
| FB | Average | 95% | 99% | Max |
| A-B | 0.005 | 0.05 | 0.07 | 0.11 |
| A-C | 0.003 | 0.03 | 0.04 | 0.06 |
| A-D | 0.003 | 0.05 | 0.07 | 0.12 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

NOTE: The MOS-LQO verification does not include testing of the EVS JBM solution. Conformance for the EVS JBM solution in [11] is FFS.

# A.4 Reference Implementations

To get the snr and mld corridor as well as the thresholds values for the MOS-LQO, as set of references implementations were used. Table A.8 list the implementations used for references, including compiler, target platform, compiler setting. These implementations are based on mainstream compilers and platforms and used the latest version of EVS code defined in TS 26.443 [8]. These implementations are not bit-exact between themselves or with the 3GPP reference implementation (Visual Studio 2018, Release mode).

Table A.8: List of Reference Implementations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Platform | Compiler | Optimization | OS |
| aarch64\_gnu-gcc-8\_-armv8\_O2 | ARMv8 | GCC v8 | O2 | Linux |
| aarch64\_gnu-gcc-8\_armv8\_O3 | ARMv8 | GCC v8 | O3 | Linux |
| clang-6\_armv8 \_O2 | ARMv8 | Clang v6 | O2 with FMA | Linux |
| clang-6\_armv8\_O3 | ARMv8 | Clang v6 | O3 with FMA | Linux |
| clang-6.0\_x86\_64\_O2 | x86\_64 | Clang v6 | O2 | Linux |
| gcc-7\_i686\_-O0 | i686 | GCC v7 | O0 | Linux |
| gcc-7\_i686\_-O1 | i686 | GCC v7 | O1 | Linux |
| gcc-7\_i686\_-O2 | i686 | GCC v7 | O2 | Linux |
| gcc-7\_i686\_-O3 | i686 | GCC v7 | O3 | Linux |
| icc-19\_x86\_64\_avx2 | x86\_64 | ICC v19 | O3 with FMA | Linux |

Annex B (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2014-09 | SA#65 | SP-140459 |  |  |  | Presented at TSG#65 for approval | 1.0.0 |
| 2014-09 | SA#65 |  |  |  |  | Approved at TSG#65 | 12.0.0 |
| 2014-12 | SA#66 | SP-140725 | 0001 |  |  | Update of existing test vectors for the fixed-point EVS codec | 12.1.0 |
| 2014-12 | SA#66 | SP-140725 | 0002 | 1 |  | Inclusion of test vectors for the floating-point EVS codec | 12.1.0 |
| 2015-03 | SA#67 | SP-150085 | 0003 |  |  | Update of test vectors for the EVS codec | 12.2.0 |
| 2015-06 | SA#68 | SP-150202 | 0004 |  |  | Update of test vectors for the EVS codec | 12.3.0 |
| 2015-09 | SA#69 | SP-150434 | 0005 | 1 |  | Update of test vectors for the EVS codec | 12.4.0 |
| 2015-12 | SA#70 | SP-150639 | 0006 |  |  | Update of test vectors for the EVS codec | 12.5.0 |
| 2015-12 | SA#70 |  |  |  |  | Version for Release 13 | 13.0.0 |
| 2016-03 | SA#71 | SP-160064 | 0008 |  |  | Update of test vectors for the EVS codec | 13.1.0 |
| 2016-06 | SA#72 | SP-160257 | 0010 |  | A | Update of test vectors for the EVS codec | 13.2.0 |
| 2016-09 | SA#73 | SP-160589 | 0012 |  | A | Update of test vectors for the EVS codec | 13.3.0 |
| 2017-03 | SA#75 |  |  |  |  | Alignment of source code and test vectors versions (update of Readme.txt file) | 13.3.1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2017-03 | 75 |  |  |  |  | Version for Release 14 | 14.0.0 |
| 2017-12 | 78 | SP-170820 | 0017 | 2 | A | Update of test vectors for the EVS codec | 14.1.0 |
| 2018-06 | 80 | SP-180261 | 0020 | - | A | Update of test vectors for the EVS codec | 14.2.0 |
| 2018-06 | 80 |  |  | - |  | Version for Release 15 | 15.0.0 |
| 2018-12 | 82 | SP-180965 | 0024 | - | A | Update of test vectors for the EVS codec | 15.1.0 |
| 2019-03 | 83 | SP-190036 | 0025 | 2 | B | Correction and addition of reference to Alt\_FX\_EVS implementation | 16.0.0 |
| 2019-12 | 86 | SP-190998 | 0027 | 5 | C | EVS Non Bit Exact Float conformance | 16.1.0 |
| 2020-03 | 87-e | SP-200104 | 0028 | 1 | F | Corrections to Floating-Point Conformance Scripts | 16.2.0 |
| 2020-03 | 87-e |  |  |  |  | Editorial | 16.2.1 |
| 2020-07 | 88-e | SP-200585 | 0034 |  | A | Update of test vectors for the EVS codec | 16.3.1 |
| 2021-09 | SA#93-e | SP-210825 | 0040 | 1 | A | Update of test vectors for the EVS codec | 16.4.0 |
| 2021-12 | SA#94-e | SP-211345 | 0045 | - | A | Update of test vectors for the EVS codec | 16.5.0 |