ETSI TS 126 448 V16.0.0 (2020-11)

Universal Mobile Telecommunications System (UMTS);
LTE;
Codec for Enhanced Voice Services (EVS);
Jitter Buffer Management
(3GPP TS 26.448 version 16.0.0 Release 16)

Reference
RTS/TSGS-0426448vg00

Keywords
LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Legal Notice

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

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

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 defines the Jitter Buffer Management solution for the Codec for Enhanced Voice Services (EVS).

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 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[4] 3GPP TS 26.071: "Mandatory speech CODEC speech processing functions; AMR speech Codec; General description".

[5] 3GPP TS 26.171: "Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description".

[6] 3GPP TS 26.442: "Codec for Enhanced Voice Services (EVS); ANSI C code (fixed-point)".

[7] 3GPP TS 26.443: "Codec for Enhanced Voice Services (EVS); ANSI C code (floating-point)".

[8] 3GPP TS 26.131: "Terminal acoustic characteristics for telephony; Requirements".

[9] IETF RFC 4867 (2007): "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", J. Sjoberg, M. Westerlund, A. Lakaniemi and Q. Xie.

[10] 3GPP TS 26.452: "Codec for Enhanced Voice Services (EVS); ANSI C code; Alternative fixed-point using updated basic operators".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$s_x(n)$ Time signal and time index n in context x, e.g. x can be inp, out, HP, pre, etc.

L_x Frame length / size of module x

E_x Energy values in context of x

C_x Correlation function in context x

3.3 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].

AMR	Adaptive Multi Rate (codec)
AMR-WB	Adaptive Multi Rate Wideband (codec)
CNG	Comfort Noise Generator
DTX	Discontinuous Transmission
EVS	Enhanced Voice Services
FB	Fullband
FIFO	First In, First Out
IP	Internet Protocol
JBM	Jitter Buffer Management
MTSI	Multimedia Telephony Service for IMS
NB	Narrowband
PCM	Pulse Code Modulation
PLC	Packet Loss Concealment
RTP	Real Time Transport Protocol
SID	Silence Insertion Descriptor
SOLA	Synchronized overlap-add
SWB	Super Wideband
TSM	Time Scale Modification
VAD	Voice Activity Detection
WB	Wideband

3.4 Mathematical Expressions

For the purposes of the present document, the following conventions apply to mathematical expressions:

$\lceil x \rceil$ indicates the smallest integer greater than or equal to x: $\lceil 1.1 \rceil = 2$, $\lceil 2.0 \rceil = 2$ and $\lceil -1.1 \rceil = -1$

$\lfloor x \rfloor$ indicates the largest integer less than or equal to x: $\lfloor 1.1 \rfloor = 1$, $\lfloor 1.0 \rfloor = 1$ and $\lfloor -1.1 \rfloor = -2$

$\min(x_0,\ldots x_{N-1})$ indicates the minimum of $x_0,\ldots, x_{N-1}$, N being the number of components

$\max(x_0,\ldots x_{N-1})$ indicates the maximum of $x_0, \ldots, x_{N-1}$

$\sum$ indicates summation

4 General

4.1 Introduction

The present document defines the Jitter Buffer Management solution for the Codec for Enhanced Voice Services (EVS) [2]. Jitter Buffers are required in packet-based communications, such as 3GPP MTSI [2], to smooth the inter-arrival jitter of incoming media packets for uninterrupted playout.

The solution is used in conjunction with the EVS decoder and can also be used for AMR [4] and AMR-WB [5]. It is optimized for the Multimedia Telephony Service for IMS (MTSI) and fulfils the requirements for delay and jitter-induced concealment operations set in [2].

The procedure of the present document is recommended for implementation in all network entities and UEs supporting the EVS codec.

The present document does not describe the ANSI C code of this procedure. For a description of the two fixed-point ANSI C code implementations, using different sets of basic operators, see [6] and [10] respectively; for a description of the floating-point ANSI C code implementation see [7].

In the case of discrepancy between the EVS Jitter Buffer Management described in the present document and its ANSI-C code specification contained in [6], the procedure defined by [6] prevails. In the case of discrepancy between the procedure described in the present document and its ANSI-C code specification contained in [7], the procedure defined by [7] prevails. In the case of discrepancy between the procedure described in the present document and its ANSI-C code specifications contained in [10] the procedure defined by [10] prevails.

4.2 Packet-based communications

In packet-based communications, packets arrive at the terminal with random jitters in their arrival time. Packets may also arrive out of order. Since the decoder expects to be fed a speech packet every 20 milliseconds to output speech samples in periodic blocks, a de-jitter buffer is required to absorb the jitter in the packet arrival time. The larger the size of the de-jitter buffer, the better its ability to absorb the jitter in the arrival time and consequently fewer late arriving packets are discarded. Voice communications is also a delay critical system and therefore it becomes essential to keep the end to end delay as low as possible so that a two way conversation can be sustained.

The defined adaptive Jitter Buffer Management (JBM) solution reflects the above mentioned trade-offs. While attempting to minimize packet losses, the JBM algorithm in the receiver also keeps track of the delay in packet delivery as a result of the buffering. The JBM solution suitably adjusts the depth of the de-jitter buffer in order to achieve the trade-off between delay and late losses.

4.3 EVS Receiver architecture overview

An EVS receiver for MTSI-based communication is built on top of the EVS Jitter Buffer Management solution. In the EVS Jitter Buffer Management solution the received EVS frames, contained in RTP packets, are depacketized and fed to the Jitter Buffer Management (JBM). The JBM smoothes the inter-arrival jitter of incoming packets for uninterrupted playout of the decoded EVS frames at the Acoustic Frontend of the terminal.

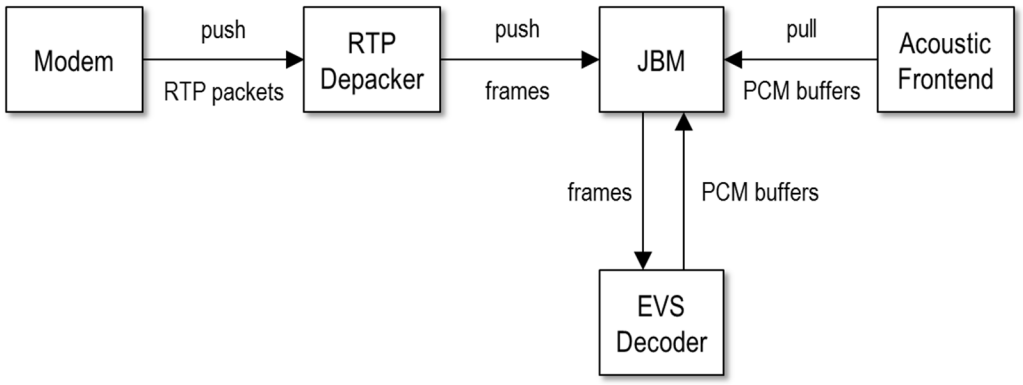

Figure 1: Receiver architecture for the EVS Jitter Buffer Management Solution

Figure 1 illustrates the architecture and data flow of the receiver side of an EVS terminal. Note that the architecture serves only as an example to outline the integration of the JBM in a terminal. This specification defines the JBM module and its interfaces to the RTP Depacker, the EVS Decoder [2], and the Acoustic Frontend [8]. The modules for Modem and Acoustic Frontend are outside the scope of the present document. The actual implementation of the RTP Depacker is outlined in a basic form; more complex depacketization scenarios depend on the usage of RTP.

Real-time implementations of this architecture typically use independent processing threads for reacting on arriving RTP packets from the modem and for requesting PCM data for the Acoustic Frontend. Arriving packets are typically handled by listening for packets received on the network socket related to the RTP session. Incoming packets are pushed into the RTP Depacker module which extracts the frames contained in an RTP packet. These frame are then pushed into the JBM where the statistics are updated and the frames are stored for later decoding and playout. The Acoustic Frontend contains the audio interface which, concurrently to the push operation of EVS frames, pulls PCM buffers from the JBM. The JBM is therefore required to provide PCM buffers, which are normally generated by decoding EVS frames by the EVS decoder or by other means to allow uninterrupted playout. Although the JBM is described for a multi-threaded architecture it does not specify thread-safe data structures due to the dependency on a particular implementation.

Note that the JBM does not directly forward frames from the RTP Depacker to the EVS decoder but instead uses frame-based adaptation to smooth the network jitter. In addition signal-based adaptation is executed on the decoded PCM buffers before they are pulled by the Acoustic Frontend. The corresponding algorithms are described in the following clauses.

5 Jitter Buffer Management

5.1 Overview

Jitter Buffer Management (JBM) includes the jitter estimation, control and jitter buffer adaptation algorithm to manage the inter-arrival jitter of the incoming packet stream. The entire solution for EVS consists of the following components:

- RTP Depacker (clause 5.2) to analyse the incoming RTP packet stream and to extract the EVS speech frames along with meta data to estimate the network jitter
- De-jitter Buffer (clause 5.6) to store the extracted EVS speech frames before decoding and to perform frame-based adaptation
- EVS decoder [1] for decoding the received EVS speech frames to PCM data
- Time-Scale Modification (clause 5.4.3) to perform signal-based adaptation for changing the playout delay
- Receiver Output Buffer (clause 5.5) to provide PCM data with a fixed frame size to the Acoustic Frontend
- Playout Delay Estimation Module (clause 5.3.5) to provide information on the current playout delay due to JBM
- Network Jitter Analysis (clause 5.3) for estimating the packet inter-arrival jitter and target playout delay
- Adaptation Control Logic (clause 5.4) to decide on actions for changing the playout delay based on the target playout delay

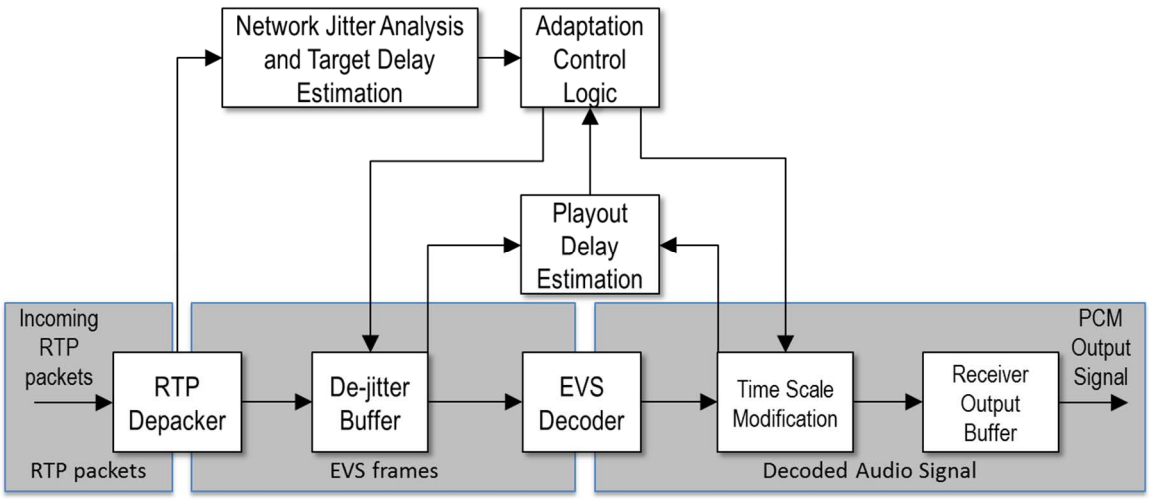

Figure 2: Modules of the EVS Jitter Buffer Management Solution

5.2 Depacketization of RTP packets (informative)

The RTP Depacker module of the JBM performs the depacketization of the incoming RTP packet stream. During this operation the EVS frames, embedded in RTP packets according to the respective RTP payload format [2], [9], are extracted and pushed to the de-jitter buffer. The RTP timestamp in an RTP packet for EVS always refers to the first EVS frame in the RTP payload. Any further EVS frames in the RTP payload are indexed in the RTP Payload Format Header by a Table of Contents (ToC) [2], [9]. The RTP Depacker performs the unpacking and calculates and assigns a media timestamp to every speech frame present in each received RTP packet.

The Jitter Buffer Management (JBM) for the EVS codec depends on information that is part of the received RTP packet stream. Each RTP packet consists of an RTP header and the RTP payload. The following data fields of the RTP header are of relevance for the JBM:

- RTP timestamp
- RTP sequence number

The marker bit in the RTP header is not evaluated by this JBM solution. Other fields in the RTP header are needed to correctly assign the incoming RTP packets to an RTP session, which is outside the scope of this specification.

All extracted frames (without NO_DATA frames) are fed to the JBM. The data structure for one frame consists of:

- Frame payload data, including the size of the payload
- Arrival timestamp of the RTP packet containing the frame
- Media timestamp in RTP timescale units, derived from the RTP timestamp of the packet
- Media duration in RTP timescale units (20 ms for EVS frames)
- RTP timescale as specified in the specification of the RTP payload format
- RTP sequence number
- SID flag
- Partial copy flag

To optimize the JBM behaviour for DTX, the JBM needs to be aware of SID frames. Determining this information depends on the implementation of the underlying audio codec. To keep the JBM independent of the audio codec, the SID flag needs to be fed to the JBM. In case of the EVS, AMR and AMR-WB codecs the SID flag can be determined from the size of the frame payload data.

Audio encoders supporting DTX typically output NO_DATA frames between SID frames to signal that a frame was not encoded because it does not contain an active signal and should be substituted with comfort noise by the audio decoder. Instead of NO_DATA frames this JBM solution uses the RTP timestamp for media time calculation. Therefore the RTP Depacker should not feed NO_DATA frames into the JBM.

The JBM handles packet reordering and duplication on the network and so the RTP Depacker can feed those frames into the JBM exactly as received, therefore a typical RTP Depacker implementation might be state-less.

5.3 Network Jitter Analysis and Delay Estimation

5.3.1 General

Estimates of the network jitter are required to control the JBM playout delay. The Jitter Buffer Management for EVS combines a short-term and a long-term jitter estimate to set the target playout delay. The playout is smoothly adapted to continuously minimize the difference between playout delay and target playout delay.

The transmission delay of a packet on the network can be seen as the sum of a fixed component (consisting of unavoidable factors such as propagation times through physical materials and minimum processing times) and a varying component (dominated by network jitter e.g. due to scheduling). As JBM does not expect synchronized clocks between the sender and receiver, the fixed delay value cannot be estimated using only the information available from the

received RTP packets. Therefore JBM ignores the fixed delay component. To estimate the varying delay component, the two basic values delay d_i and offset o_i are calculated using the arrival timestamp r_i and the media timestamp t_i. This is done for every received frame, where i relates to the current frame (most recently received), and $i-1$ relates to the previously received frame. Note that for the first received frame of a session no delay value will be calculated (d_0 is set to 0).

$$d_i = (r_i - r_{i-1}) - (t_i - t_{i-1}) + d_{i-1} \quad (1)$$

Delay calculations are done in millisecond units. A sliding window stores the offset o_i and delay values d_i for received frames.

$$o_i = r_i - t_i \quad (2)$$

The difference of the stored maximum and minimum delay values is used as an estimate for the varying delay component (network jitter). Both the target playout delay and the playout delay calculations are based on the minimum stored offset, i.e. the offset calculated for the frame received with the lowest transmission delay relative to all frames currently contained in the sliding window. The details on how this approach is used are described in the following sections.

5.3.2 Long-term Jitter

To calculate the long term jitter j_i, an array of network statistics entries (FIFO queue) is used. For each frame received in an RTP packet on the network an entry will be added to the array. An entry contains three values: delay d_i, offset o_i and RTP timestamp t_i. The time span stored in the FIFO might be different to the number of stored entries if DTX is used, for that reason the window size of the FIFO is limited in two ways. It may contain at most 500 entries (equal to 10 seconds at 50 packets per second) and at most a time span (RTP timestamp difference between newest and oldest frame) of 10 seconds. If more entries need to be stored, the oldest entry is removed. Note that in the following operations the arrays are assumed to contain only valid entries, i.e. non-arriving packets are left out from the operations when e.g. calculating minimum or maximum values.

The long term jitter j_i for the i-th received frame is calculated as the difference between the maximum delay value currently stored in the array and the minimum delay value:

$$j_i = \max(d_{i-500},...,d_i) - \min(d_{i-500},...,d_i) \quad (3)$$

5.3.3 Short-term jitter

The short term jitter estimation is done in two steps.

In the first step, the same jitter calculation is used as for long term estimation with the following modifications: The window size of a first array ("Fifo1") is limited to a maximum of 50 entries and a maximum time span of one second. The resulting temporary jitter value k_i is calculated as the difference between the 94 % percentile delay value d_{p94} currently stored in the first array and the minimum delay value of the first array:

$$k_i = d_{p94} - \min(d_{i-50},...,d_i) \quad (4)$$

In the second step, first the different offsets between the short term and long term FIFOs are compensated for this result to generate another temporary jitter value l_i, i.e. by adding the difference of the minimum offset value stored in Fifo1 and the minimum offset value stored in longTermFifo to k_i:

$$l_i = k_i + \min(o_{i-50},...,o_i) - \min(o_{i-500},...,o_i) \quad (5)$$

This result is added to a second array ("Fifo2") containing the temporary jitter values l_i with a maximum window size of 200 entries and a maximum time span of four seconds. Finally, the maximum jitter value of the Fifo2 is rounded up to an integer multiple of the frame size and used as the short term jitter m_i:

$$m_i = \lceil \max(l_{i-200},...,l_i) / 20ms \rceil * 20ms \quad (6)$$

5.3.4 Target Playout Delay

The target playout delay is different for active frames, inactive periods and for the first frame after inactivity.

To calculate the target playout delay for active frames, the long term and short term jitter estimations are combined to set a lower threshold (minimum target playout delay) u_i and upper threshold (maximum target playout delay) v_i. The target playout delay calculation is configured with (extra redundant frame delay) $g = 0$ and (delay reserve) $h = 15$. These variables are modified in case of partial redundancy as described in subclause 6.3.

$$v_i = m_i + 60ms + g \tag{7}$$

$$u_i = \min(j_i + 20ms + g + h, v_i) \tag{8}$$

For active frames the signal-based adaptation is executed if the playout delay exceeds either threshold.

In case of DTX the frame-based adaptation is executed by inserting or removing NO_DATA frames to increase or reduce the playout delay to follow the DTX target playout delay w_i, which is set to the minimum of the long-term and short-term jitter:

$$w_i = \min(j_i + h, m_i) \tag{9}$$

For the first active frame after DTX, frame-based adaptation is executed by inserting NO_DATA frames to increase the playout delay to match the target playout delay for the first active frame z_i, which is set to the average of the minimum and maximum threshold for active frames.

$$z_i = (u_i + v_i + \frac{h}{4}) / 2 \tag{10}$$

5.3.5 Playout Delay Estimation

The playout delay is calculated whenever the decoder is executed, regardless of whether it was executed to decode a received frame, conceal a missing frame or generate comfort noise during DTX. The previous playout delay p_{k-1} specifies the buffering delay between the previous playout time $k-1$ when a frame was played and the time that frame could have been received were its transmission delay equal to the lowest of all frames currently contained in the long-term FIFO.

$$p_{k-1} = q_{k-1} - \min(o_{i-500}, ..., o_i) + b_k \tag{11}$$

The variable b_k describes the duration of the samples buffered in the Receiver Output Buffer module at playout time k and $\min(o_{i-500}, ..., o_i)$ is the minimum offset value stored in the long-term FIFO, irrespective of which frame is currently played out. The previous playout offset q_{k-1} is recalculated whenever a received frame is popped from the De-jitter Buffer module using the current system time s_{k-1} and the RTP timestamp of the frame that was popped.

$$q_{k-1} = s_{k-1} - t_{k-1} \tag{12}$$

Since q_{k-1} uses the RTP timestamp, it cannot be calculated for concealed frames or comfort noise and thus for these cases it needs to be estimated. q_k is assumed to be identical to the previous value q_{k-1}. In addition, q_k shall be updated whenever frame-based adaptation occurs. The duration of the inserted frame is added and the duration of the deleted frame is subtracted from q_k, respectively.

5.4 Adaptation Control Logic

5.4.1 Control Logic

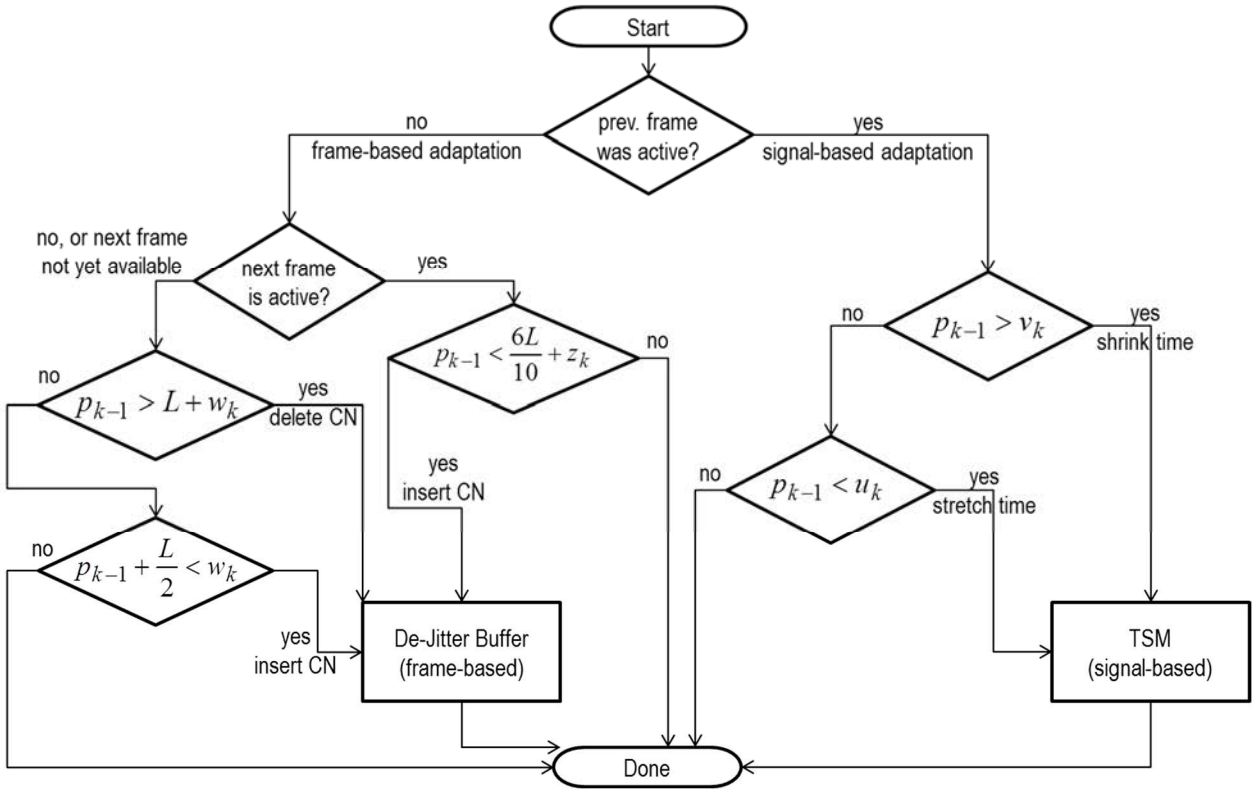

Figure 3: Control Logic of the EVS Jitter Buffer Management Solution

The Control Logic module compares the playout delay with the target playout delay as outlined in Figure 3. If the difference exceeds a threshold, it triggers adaptation to modify the playout delay and therefore also the depth of the De-Jitter Buffer. Different types of adaptation are supported and selected in a signal-adaptive manner. During comfort noise (SID-flag is active), frame-based adaptation will be triggered and executed by the De-Jitter Buffer module. During active periods, signal-based adaptation is triggered and executed by the Time Scale Modification module.

An initialization phase is used at startup. The JBM solution outputs audio samples before receiving and decoding the first frame which is received with some network delay. During this initialization phase the audio decoder is not used yet and PCM buffers are returned filled with zero samples.

5.4.2 Frame-based adaptation

5.4.2.1 General

Adaptation on the frame level is performed on coded speech frames, i.e. with a granularity of one speech frame of 20 ms duration. Inserting or deleting speech frames results in adaptation with higher distortion but allows faster buffer adaptation than signal-based adaptation. Inserting or deleting NO_DATA frames during DTX allows fast adaption while minimizing distortion.

5.4.2.2 Insertion of Concealed Frames

If the de-jitter buffer runs empty because the next frame has not yet been received during a delay spike, a missing frame is signalled to the EVS decoder in order to create samples for playout. The PLC module integrated in the decoder is used to extrapolate artificial samples. Subsequently arriving speech frames are then again fed to the EVS decoder, including the late arriving frame that was initially replaced by PLC.

5.4.2.3 Frame Dropping

The De-Jitter Buffer has a fixed capacity to avoid excessive delay and memory usage. For some corner cases like major delay bursts on the network, the De-Jitter Buffer might still overflow. In that case the oldest frame (lowest timestamp) will be dropped by removing it from the De-Jitter Buffer until one slot is free to store the newest received frame.

In addition, frames can also be dropped if they are received out-of-order as a result of packet reordering on the network. In most cases the De-Jitter Buffer will undo the reordering before frames are pushed into the audio decoder. If a reordered frame was received too late to be decoded in order because the frame with the following timestamp was already fed into the audio decoder, it is considered to be no longer of any use to the decoder and the De-Jitter Buffer will drop such a frame instead of storing it.

The insertion of concealed frames as described in subclause 5.4.2.2 increases the playout delay. To avoid excessive build-up of delay, the first frame to be decoded after the insertion of concealed frames will be dropped if the playout delay after playing this late frame would be higher than the target playout delay.

5.4.2.4 Comfort Noise Insertion in DTX

During DTX the playout delay is increased by pushing an additional NO_DATA frame into the decoder to insert an additional frame containing comfort noise. The first active frame after DTX is decoded after additional comfort noise frames are decoded for frame-based adaptation. Adaptation by insertion of NO_DATA frames between two SID frames or between an SID frame and the first active frame results in negligible distortion of the signal.

5.4.2.5 Comfort Noise Deletion in DTX

Similar to comfort noise insertion in DTX, comfort noise frames can also be deleted by omitting the decoding of NO_DATA frames. Comfort noise is deleted to control the playout delay between SID frames and before the first active frame after DTX.

5.4.3 Signal-based adaptation

5.4.3.1 General

To alter the playout delay the output signal of the decoder is time-warped. The time-warping is performed by the Time Scale Modification module, which generates additional samples for increasing the playout delay or which removes samples from the output signal to reduce the playout delay.

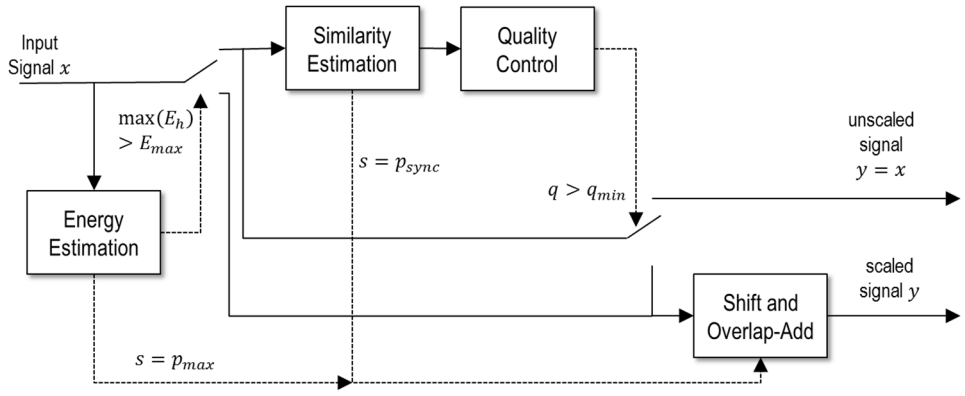

Figure 4: Signal-based adaptation based on signal characteristics

A SOLA (synchronized overlap-add) algorithm with built-in quality control is used for adaptation by performing time-scale modification of the signal without altering the pitch. The level of adaptation is signal-dependent, as also outlined in Figure 4:

- signals that are classified by the quality-control as "introducing severe distortions when scaled" are never scaled;
- low-level signals, which are close to silent, are time-scaled without synchronization to the maximum possible extent;

- signals that are classified as "time-scalable", e.g. periodic signals, are scaled by a synchronized overlap-add of a shifted representation of the input signal with the input signal to the Time Scale Modification module. The shift itself is derived from a similarity measure. The shift differs depending on whether the signal is to be shortened or lengthened.

The time-scale modification algorithm shifts a portion of samples to a position that yields synchronization. The original signal is cross-faded with the shifted portion using overlap-add, followed by the remaining samples that follow the shifted portion. The level of time-scaling is signal-dependent.

To yield synchronization the shifted portion and the original frame are cross-correlated using a complexity-reduced normalized cross correlation function. The original frame size as fixed by the EVS decoder is 20ms, i.e. 640 samples for SWB. The size of the template is 10ms, i.e. 320 samples for SWB. The general parameters of the time-scale modification are listed in Table 1.

Table 1: Time-scale Modification Parameters

		8 kHz	16 kHz	32 kHz	48 kHz
L	frame size [samples]	160	320	640	960
L_{seg}	segment size [samples]	80	160	320	480
p_{min}	minimum pitch	20	40	80	120
l_{search}	search length	100	200	400	600
p_{max}	maximum pitch	120	240	480	720

5.4.3.2 Time-shrinking

Time-shrinking reduces the size L of a frame to L_{out}. L_{out} will differ to L by 2,5-10 ms, i.e. the resulting frame duration will be in the range 10-17,5 ms. The amount of time-scaling depends on the position p_{sync} of the best matching candidate segment that yields highest similarity to the first segment a of the input signal. The start position s_{start} and end position s_{end} used to search for the candidate segment inside the input frame x for time-shrinking are listed in Table 2.

Table 2: Time-shrinking Parameters

		8 kHz	16 kHz	32 kHz	48 kHz
s_{start}	start search position [samples]	20	40	80	120
s_{end}	end search position [samples]	80	160	320	480

The output frame y is the result of a cross-fade of the first segment a of the input frame and the best matching candidate segment, which is shifted by $s = p_{sync}$. Samples following the candidate segment are appended to the merged signal to yield continuity with following frames.

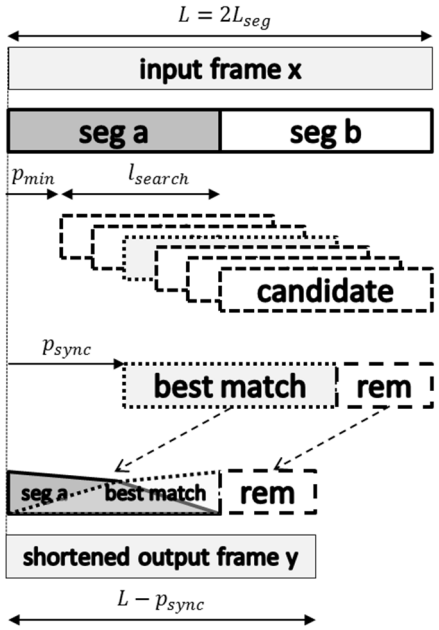

Figure 5: Shortening of an input frame

Note that the time-shrinking algorithm does not need any look-ahead samples and is therefore not introducing extra delay, however an extra buffer is used to generate output frames y' with a fixed frame length from the variable length output frames y (clause 5.5).

5.4.3.3 Time-stretching

Time-stretching increases the size L of an input frame x to L_{out} for the output frame y . L_{out} will differ to L by 2,5-15 ms, i.e. the resulting frame duration will be in the range 22,5-35 ms. The amount of time-scaling depends on the position p_{sync} of the best matching candidate segment that yields highest similarity to the first segment a of the input signal. The start position s_{start} and end position s_{end} used to search for the candidate segment inside the input frame x for time-stretching are listed in Table 3.

The preceding input frame is taken to search for positions with similarity to the first segment of the current input frame.

Table 3: Time-stretching Parameters

		8 kHz	16 kHz	32 kHz	48 kHz
s_{start}	start search position [samples]	-120	-240	-480	-720
s_{end}	end search position [samples]	-20	-40	-80	-120

The output frame y is the result of a cross-fade of the first segment a of the input frame and the best matching candidate segment, which is shifted by $s = p_{sync}$. Note that p_{sync} is bounded by s_{start} and s_{end} and is therefore a negative number. Samples following the candidate segment are appended to the merged signal to yield continuity with following frames.

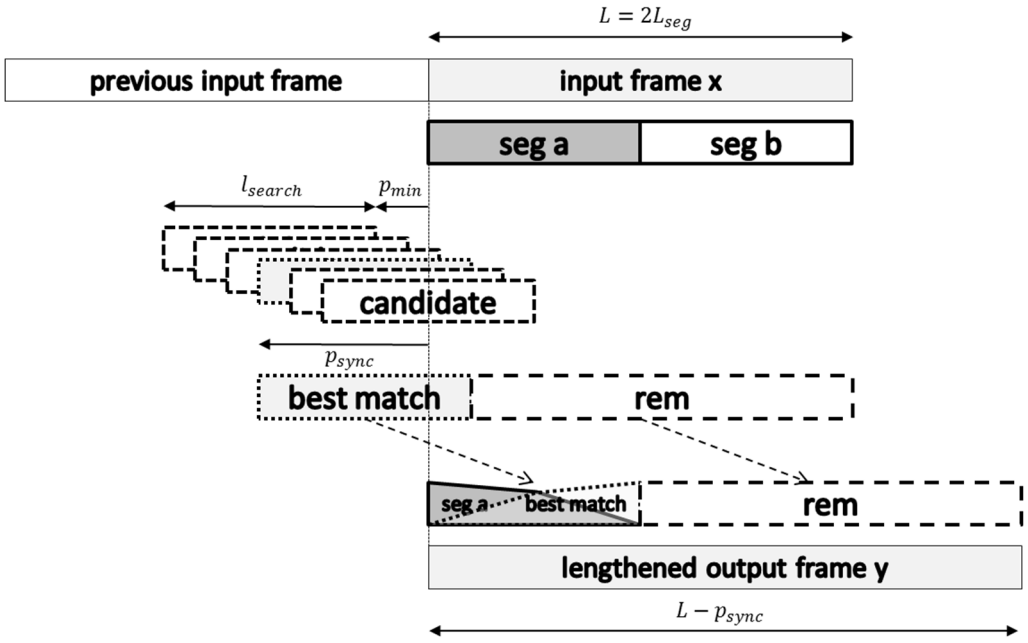

Figure 6: Lengthening of an input frame

Note that the time-stretching algorithm does not need any look-ahead samples but requires the previous frame to be available. Therefore no extra delay is introduced, however an extra buffer is used to generate output frames y' with a fixed frame length from the variable length output frames y (clause 5.5).

5.4.3.4 Energy Estimation

Low-level signals are detected by analysing 1ms subsegments h of the input signal x, including the previous frame. If all subsegment energies E_h of the subsegments to be merged are below a threshold E_{max}, with $E_{max} = -65dB$, then the frame is scaled to the maximum extent:

- a 20 ms frame is shortened to 10ms by setting $s = s_{end}$ for time-shrinking according to Table 2
- a 20 ms frame is extended to 35ms by setting $s = s_{start}$ for time-stretching according to Table 3

The generation of the output frame y is performed by a cross-fade of the first segment a of the input frame and the by s shifted segment, followed by the remaining samples. Note that for low-level signals the output frame is generated without similarity estimation or quality estimation.

5.4.3.5 Similarity Measurement

To estimate similarity a non-normalized cross-correlation of the template segment with other positions of the template in the signal is maximized. In order to limit the computational complexity, a $1:o$ subsampled signal is used to estimate the correlation value, i.e. only every o^{th} sample is used. The subsampling parameters are set as listed in Table 4.

Table 4: Similarity Measurement complexity Parameters

		8 kHz	1 6kHz	32 kHz	48 kHz
o	signal subsampling	1	2	4	6
m	correlation subsampling	1	1	2	3

$$C_\sigma = \sum_{n=1}^{l\_seg/o} x(n*o)x(n*o+\sigma) \tag{13}$$

A hierarchical search for the best match is performed that initially skips every *m*-th offset, i.e. the search for the maximum correlation is performed on a subsampled set of correlation values, with m as listed in Table 4. The offset with highest correlation is then used as starting point for a hierarchical search where $m = \lfloor m*0.5 \rfloor$ and $l_{search} = \lfloor l_{search} * 0.5 \rfloor$ after each step. s_{start} is set to $s_{start} = p_{sync} - \lfloor 0.5 l_{search} \rfloor$ for narrowing the search range. This hierarchical search is performed until $m = 1$.

5.4.3.6 Quality Control

After similarity estimation a quality measure q is calculated. To calculate q the normalized cross-correlation in (14) is used.

$$C_\tau = \frac{\sum_{n=0}^{L_{seg}} x(n)x(n+\tau)}{\sqrt{\sum_{n=0}^{L_{seg}} x^2(n) + \sum_{n=0}^{L_{seg}} x^2(n+\tau)}} \tag{14}$$

C_τ is evaluated with up to four different values of τ, with τ set to p_{sync}, $2p_{sync}$, $\frac{1}{2}p_{sync}$, and $\frac{3}{2}p_{sync}$. q is then the sum of the products of the correlations in (15)

$$q = C_{p\_sync} C_{2*p\_sync} + C_{3/2*p\_sync} C_{1/2*p\_sync} \tag{15}$$

In the event that one of the calculations needs to access samples that are not available in the current or the previous frame, the values are replaced with C_{p_sync} using the following scheme:

$$C_{2*p\_sync} = C_{p\_sync} \tag{16}$$

$$C_{3/2*p\_sync} = C_{p\_sync} \tag{17}$$

$$C_{1/2*p\_sync} = C_{p\_sync} \tag{18}$$

The quality q is then used in the main scaling operations 5.4.3.2 and 5.4.3.3 to decide whether the scaling operation is performed or whether the scaling is deferred to a subsequent frame.

Positive values indicate periodicity, whereas negative values indicate that the scaling could produce a perceptually distorted signal. The threshold for frame scaling is $q = 1.0$ initially, however the threshold is dynamically adapted depending on the number of successive scaled or non-scaled frames. For each scaled frame the threshold is increased by 0,2 to avoid too many subsequent scaling operations. For each non-scaled frame the threshold is lowered by 0,1 to enable time-scaling also for less periodic signals.

For multi-channel operation the channel with the highest sum of energies is used for determining the quality of the scaling operation for all channels. Therefore the denominator from Formula (14) is summed up to determine the channel of highest sum of energies during quality estimation.

5.4.3.7 Overlap-add

Finally, if the time-scaling is performed, the output signal $y(n)$ is constructed by an overlap-add of the first segment of the input signal $x(n)$ with the shifted input signal $x(n+s)$. Note that s is negative for expansion of the signal.

$$y(n) = x(n)(1-w(n)) + x(n+s)w(n), \quad n = 1,...,L_{seg} \tag{19}$$

The overlap-add is performed using a cos-shaped Hann window. Note that the second half of the window is not used.

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{L-1}\right)\right) \quad (20)$$

The remaining samples of the shifted signal up to the most recent sample are then appended. The total length of the output signal is $L_{out} = L - s$.

$$y(n) = x(n+s), \quad n = L_{seg} + 1, ..., L - s \quad (21)$$

5.5 Receiver Output Buffer

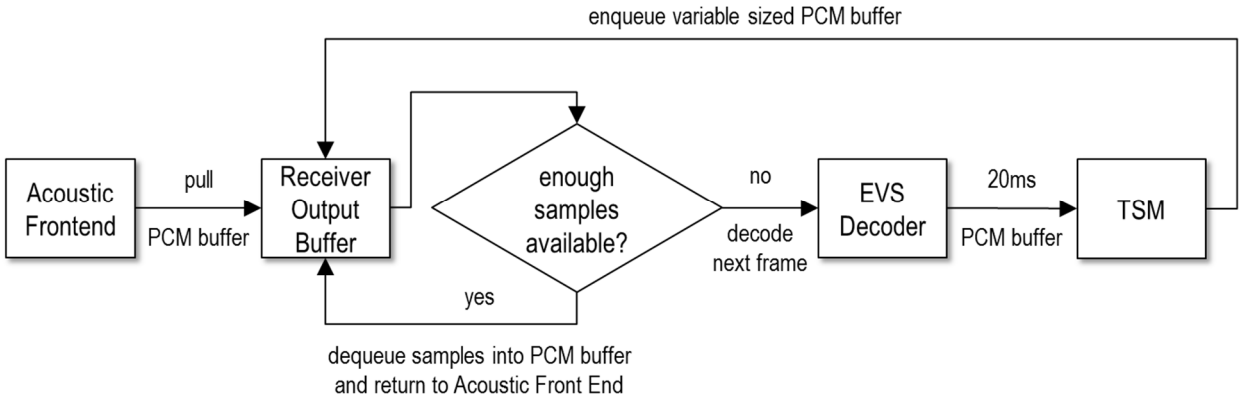

Figure 7: Receiver Output Buffer

A FIFO queue for PCM data called the Receiver Output Buffer is present in the EVS JBM solution. Since the Time Scale Modification module does not generate speech frames of length 20 ms if signal-based adaptation has occurred, this buffer ensures that the Acoustic Frontend is provided with PCM data with a fixed size of 20 ms. As outlined in Figure 7, the Receiver Output Buffer uses a ringbuffer data structure where the variable sized output of the Time Scale Modification module is enqueued and a 20ms buffer is dequeued for the Acoustic Frontend.

A FIFO queue for PCM data called the Receiver Output Buffer is present in the EVS JBM solution. As outlined in Figure 7 it provides PCM data with a fixed size of 20 ms for the Acoustic Frontend, although the Time Scale Modification module does not generate speech frames of length 20 ms if signal-based adaptation has occurred. The Receiver Output Buffer uses a ringbuffer data structure where the variable sized output of the Time Scale Modification module is enqueued and a 20 ms buffer is dequeued for the Acoustic Frontend.

5.6 De-Jitter Buffer

RTP packets are transmitted on the network with a time-varying delay (network jitter) and might be reordered, lost or duplicated. The De-Jitter Buffer stores the frames contained in RTP packets received from the network and prepares them to be fed into the audio decoder in the correct order.

The De-Jitter Buffer uses a ringbuffer data structure with a fixed capacity. To avoid excessive delay and memory usage in extreme cases like major delay bursts on the network, the ringbuffer memory is allocated at initialization with a capacity to store up to three seconds of active audio data, i.e. 150 entries for a frame duration of 20 ms. In case of an overflow of the De-Jitter Buffer the oldest frames (lowest timestamp) will be dropped from the ringbuffer.

The ringbuffer is sorted by the media timestamp in ascending order. To forward a frame from the De-Jitter Buffer to the audio decoder, the frame stored at the beginning of the ringbuffer is dequeued, i.e. the frame with the lowest timestamp. To enqueue a frame the De-Jitter Buffer uses a binary search algorithm to compare the timestamps, in order to insert the new frame at the correct position in the ringbuffer and undo any potential reordering that occurred on the network. To handle packet duplication on the network, a newly received frame will be ignored if another frame with equal timestamp and size is already stored in the ringbuffer. If a newly received frame has the same timestamp but different size than an existing frame, the frame with the greater buffer size will be stored in the ringbuffer and the other frame will be ignored, or dropped if already stored. This allows for the sending of redundant frames (either identical copies of a frame or frames where the same signal is encoded with a lower bitrate mode) as forward error correction to combat high packet loss on the network.

The De-Jitter Buffer does not explicitly handle missing frames, e.g. frames which are lost on the network or are not yet received, and thus does not store NO_DATA frames. Instead, the Adaptation Control Logic module is responsible to decide if a frame should be dequeued from the De-Jitter Buffer. The Adaptation Control Logic module will preview the first frame stored in the De-Jitter Buffer and compare its timestamp with the expected timestamp for playout before dequeuing the frame. The depth of the De-Jitter Buffer is therefore dynamic and controlled by the Adaptation Control Logic module.

6 Decoder interaction

6.1 General

This JBM solution has been optimized for the EVS codec but is also intended to be used with the existing AMR and AMR-WB codecs. Partial redundancy is only supported by the RTP Depacker for EVS frames, the interaction is defined below.

6.2 Decoder Requirements

The defined JBM depends on the decoder processing function to create an uncompressed PCM frame from a coded frame. The JBM requires that the number of channels and sampling rate in use is known in order to initialize the signal-based adaptation. The JBM also requires the presence of PLC functionality to create a PCM frame on demand without a coded frame being available as input for the decoder for missing or lost frames.

The JBM will make use of DTX for playout adaptation during inactive periods when noise is generated by the CNG. It has however its own functionality integrated for playout adaptation of active signals if the codec does not currently use or support DTX, as well as for adaptation during a long period of active signal. To use DTX the RTP Depacker needs to determine if a frame is an SID frame or an active frame and provide that information to the JBM together with the respective frame. During DTX the JBM may alter the periods between SID frames or between the last SID frame and the first active frame to use the CNG of the decoder to create additional PCM buffers with comfort noise or to omit comfort noise frames.

The JBM expects that the decoder outputs PCM frames with 20ms duration and a fixed audio sampling rate set at initialization. If the codec supports bandwidth switching, a resampling functionality is required in the decoder to provide PCM frames at the set sampling rate.

6.3 Partial Redundancy

The EVS channel aware mode [2] uses partial redundant frames that include just a subset of parameters that are most critical for decoding or arresting error propagation.

In this mode redundancy is transmitted in-band as part of the codec payload as opposed to transmitting redundancy at the transport layer (e.g. by including multiple packets in a single RTP payload). Including the redundancy in-band allows the transmission of redundancy to be either channel controlled (e.g. to combat network congestion) or source controlled. In the latter case, the encoder can use properties of the input source signal to determine which frames are most critical for high quality reconstruction at the decoder and selectively transmit redundancy for those frames only. Another advantage of in-band redundancy is that source control can be used to determine which input frames can best be coded at a reduced frame rate in order to accommodate the attachment of redundancy without altering the total packet size. In this way, the channel aware mode includes redundancy in a constant-bit-rate channel (13,2 kbps).

The JBM inspects incoming frames at 13,2 kbps to parse the flag if the frame contains a partial copy and the FEC offset of the copy. The frame is stored in the De-Jitter Buffer with the partial copy flag set to false to signal that the primary copy in the frame payload data should be decoded. If the frame contains a partial copy, then in addition a copy of the frame is stored in the De-Jitter Buffer with modified metadata:

- the partial copy flag is set to true to signal to the EVS decoder to use the partial copy stored in the frame payload data;
- the media timestamp is set to the media timestamp of the primary copy frame minus the frame duration multiplied by the FEC offset of the partial copy;

- the SID flag is set to false.

The De-Jitter Buffer stores only one frame per media timestamp. If a newly received primary copy frame is to be stored and a partial copy frame with same media timestamp is already available, the partial copy frame will be replaced by the primary copy frame. When a frame is dequeued from the De-Jitter Buffer for decoding, the EVS decoder uses the partial copy flag to decide between decoding the primary copy or using the partial copy to improve PLC for the unavailable primary copy.

Partial copies are typically received with higher delay than primary copies because they are delayed by the FEC offset and packed and sent together with the later primary copy. Both primary and partial copies are used for frame-based adaptation. If the De-Jitter Buffer runs empty, concealed frames are inserted for decoding. Both primary and partial copies that arrive after the corresponding frames were already concealed, will be stored in the De-Jitter Buffer and used for playout of the next frame. This method that increases the playout delay is also described in subclause 5.4.2.2. Also the frame dropping method described in subclause 5.4.3.2 is still applied for primary copies. In addition, to avoid excessive build-up of delay by late partial copies, they will be dropped instead of stored in the De-Jitter Buffer if the playout delay after playing the late partial copy would increase by more than 40ms (offset 5) or 80ms (offset 7). For offset 2 and 3 partial copies received for frames, which were already concealed, are always dropped. Instead of frame-based time-stretching, the JBM uses a modified jitter estimation to consider the delay of partial copies with low offsets in the target playout delay calculation. If a partial copy with offset 2 or 3 is useful (primary copy lost and partial copy received in time), it is considered in the long-term jitter estimation (subclause 5.3.2) by adding an entry with the network statistics of the partial copy to the array. In addition to those changes, also the target playout delay calculation (subclause 5.3.4) is modified to use $h = 0$.The following settings for partial redundancy with high offsets apply: $g = 100$ if offset 5 is active and $g = 140$ for offset 7. For offset 2 and 3 the target playout delay is calculated with the default value $g = 0$.

As described in subclause 5.8.3 of [2], the optimal partial redundancy offset and frame erasure rate indicator are two receiver feedback parameters which may be used to inform the channel aware mode encoder of the channel characteristics at the receiver. The following section describes the mechanism to compute these parameters at the receiver. It is noted that in the absence of such a feedback mechanism, the channel aware mode encoder will operate using the default setting as specified in subclause 5.8.3 of [2].

6.3.1 Computation of the Partial Redundancy Offset

The difference in time units between the transmit time of the primary copy of a frame (n) and the transmit time of the redundant copy of that frame which is piggy backed onto a future frame $(n + X)$) is called the FEC offset X . The optimal FEC offset is the value which maximizes the probability of availability of a partial redundancy copy when there is a frame loss. This however depends on the delay jitter and the loss statistics of the channel which is available at the JBM. Hence, the JBM periodically computes the optimal FEC offset. To enable channel condition based FEC offset adaptation at the encoder, the FEC offset computed at the receiver is communicated to the EVS encoder via a receiver feedback mechanism. This closed loop FEC offset control strategy will ensure the optimal use of redundant information and consequently improve the end user experience. The following section explains how the optimal FEC offset is periodically computed at the receiver JBM.

Let $P(x \mid H, K)$ be the probability mass function of successful recovery of partial frame for the FEC offset x , given the JBM statistics of past H frames including all lost frames K . Also $\Phi_k(x)$ is the indicator function such that:

$$\Phi_k(x) = \begin{cases} 1; & \textit{if frame } k + x \textit{ is available in the JBM} \\ 0; & \textit{if frame } k + x \textit{ is not available in the JBM} \end{cases} \tag{22}$$

Now we can find the probability mass function $P(x \mid H, K)$ as follows,

$$P(x \mid H, K) = \frac{\sum_{k=1}^{K} \Phi_k(x)}{\sum_{q=1}^{D} \sum_{k=1}^{K} \Phi_k(q)} \tag{23}$$

Note that D is the maximum possible JBM depth, which is also the maximum possible FEC offset. The denominator of the above equation is a normalization term so we can simply compute $\sum_{k=1}^{K}\Phi_k(x)$ for all possible offsets x and pick the optimum FEC offset X_{opt} which maximizes the recovery of partial redundant information. *i.e.*

$$X_{opt} = \underset{x}{\arg\max}\left(P(x \mid H, K)\right) = \underset{x}{\arg\max}\sum_{k=1}^{K}\Phi_k(x) \tag{24}$$

The optimal FEC offset computation is carried out periodically once the total number of frames or the total number of lost frames exceeds corresponding threshold values. The optimal offset is rounded to a subset of values to facilitate the quantization using a limited number of bits. The rounded offset value is transmitted to the EVS encoder, thus the encoder can use the best offset to ensure the improved performance under varying channel conditions.

6.3.2 Computation of a frame erasure rate indicator to control the frequency of the Partial Redundancy transmission

To enable different levels of partial redundancy information transmission, the JBM periodically computes the effective frame loss rate at the input to the decoder. If the FER rate is below a predefined threshold (5 % by default), the frame erasure rate indicator is set to LO (low), otherwise it is set to HI (high).

As described in subclause 5.8.3 of [2], this parameter at the encoder controls the threshold used to determine whether a particular frame is critical or not. Such an adjustment of the criticality threshold is used to control the frequency of partial copy transmission at the encoder. The HI setting adjusts the criticality threshold to classify more frames as critical to transmit as compared to the LO setting.

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2014-09	65	SP-140463			Presented at TSG SA#65 for approval		1.0.0
2014-09	65				Appproved at TSG SA#65	1.0.0	12.0.0
2014-12	66	SP-140728	0001		Corrections	12.0.0	12.1.0
2015-12	70				Version for Release 13	12.1.0	13.0.0

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-03	75					Version for Release 14	14.0.0
2018-06	80					Version for Release 15	15.0.0
2019-03	83	SP-190036	0002	-	B	Correction and addition of reference to ALT_FX_EVS implementation	16.0.0

History

Document history		
V16.0.0	November 2020	Publication