6.8 AMR-WB-interoperable modes

6.8.1 Decoding and speech synthesis

6.8.1.1 Excitation decoding

The decoding process is performed in the following order:

Decoding of LP filter parameters: The received indices of ISP quantization are used to reconstruct the quantized ISP vector. The interpolation described in subclause 5.7.2.6 is performed to obtain 4 interpolated ISP vectors (corresponding to 4 subframes). For each subframe, the interpolated ISP vector is converted to LP filter coefficient domain a_k, which is used for synthesizing the reconstructed speech in the subframe.

The following steps are repeated for each subframe:

1. **Decoding of the adaptive codebook vector:** The received pitch index (adaptive codebook index) is used to find the integer and fractional parts of the pitch lag. The adaptive codebook vector $v(n)$ is found by interpolating the past excitation $u(n)$ (at the pitch delay) using the FIR filter described in subclause 5.7. The received adaptive filter index is used to find out whether the filtered adaptive codebook is $v1(n) = v(n)$ or $v2(n) = 0.18v(n) + 0.64v(n-1) + 0.18v(n-2)$.

2. **Decoding of the innovative vector:** The received algebraic codebook index is used to extract the positions and amplitudes (signs) of the excitation pulses and to find the algebraic codevector $c(n)$. If the integer part of the pitch lag is less than the subframe size 64, the pitch sharpening procedure is applied which translates into modifying $c(n)$ by filtering it through the adaptive prefilter $F(z)$ which consists of two parts: a periodicity enhancement part $1/(1-0.85z^{-T})$ and a tilt part $(1-\beta_1 z^{-1})$, where T is the integer part of the pitch lag and $\beta_1(n)$ is related to the voicing of the previous subframe and is bounded by [0.0,0.5].

3. **Decoding of the adaptive and innovative codebook gains:** The received index gives the fixed codebook gain correction factor $\hat{\gamma}$. The estimated fixed codebook gain g'_c is found as described in subclause 5.8. First, the predicted energy for every subframe n is found by

$$\tilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i) \qquad (1994)$$

   and then the mean innovation energy is found by

$$E_i = 10\log\left(\frac{1}{N}\sum_{i=0}^{N-1} c^2(i)\right) \qquad (1995)$$

   The predicted gain g'_c is found by

$$g'_c = 10^{0.05(\tilde{E}(n)+\bar{E}-E_i)} \qquad (1996)$$

   The quantized fixed codebook gain is given by

$$\hat{g}_c = \hat{\gamma} g'_c. \qquad (1997)$$

4. **Computing the reconstructed speech:** The following steps are for *n* = 0, ..., 63. The total excitation is constructed by:

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \qquad (1998)$$

6.8.1.2 Excitation post-processing

Before the synthesis, a post-processing of the excitation signal is performed to form the updated excitation signal, $u(n)$, as follows.

6.8.1.2.1 Anti-sparseness processing

This is the same as described in subclause 6.1.1.3.1

6.8.1.2.2 Gain smoothing for noise enhancement

This is the same as described in subclause 6.1.1.3.2

6.8.1.2.3 Pitch enhancer

This is the same as described in subclause 6.1.1.3.3.

6.8.1.3 Synthesis filtering

Once the excitation post-processing is done, the modified excitation is passed through the synthesis filter, as described in subclause 6.1.3, to obtain the decoded synthesis for the current frame. Based on the content bandwidth in the decoded synthesis signal, an output mode is determined (e.g., NB or WB). If the output mode is determined to be NB, then the content above 4 kHz is attenuated using CLDFB synthesis (e.g., as described in clause 6.9.3) and, subsequently, high frequency synthesis (6.8.3) is not performed on the bandlimited content.

6.8.1.4 Music and Unvoiced/inactive Post-processing

6.8.1.4.1 Music post processing

Most of the music post processing is the same as in as clause 6.1.1.3.4. The main difference related to the fact that a first synthesis is computed and a first stage classification is derived from this synthesis as described in subclause 5.3.1 of [6]. If the synthesis is classified as unvoiced or the content is INACTIVE (VAD ==0) or the long term background noise (E_{lt_noise}) as defined below is greater or equal to 15 dB, the AMR-IO decoder will go through the unvoiced, inactive post processing path as described in subclause 6.8.1.1.5.

The long term background noise energy is updated in case of INACTIVE frame as:

$$E_{lt\_noise} = 0.9 \cdot E_{lt\_noise} + 0.1 \cdot \left(10\log_{10}\left(\frac{1}{T'} \sum_{n=0}^{T'-1} \hat{s}^2 (L - T' + n) \right) \right) \qquad (1999)$$

and E_{lt_noise} is the long-term background noise energy. E_{lt_noise} is updated only when a current frame is classified as INACTIVE. The pitch lag value, T’, over which the background noise energy, E_{lt_noise} , is given by

$$\begin{aligned} &p = \text{round}(0.5 d_{fr}^{[2]} + 0.5 d_{fr}^{[3]}) \\ &T' = p \quad \text{if} \quad p \ge L_{subfr} \\ &T' = 2p \quad \text{if} \quad p < L_{subfr} \end{aligned} \qquad (2000)$$

where $d_{fr}^{[i]}$ is the fractional pitch lag at subframe i, L is the frame length and L_{subfr} is the subframe length. Otherwise it enters the music post processing is entered as described below.

6.8.1.4.1.1 Excitation buffering and extrapolation

This is the same as described in subclause 6.1.1.3.4.1

6.8.1.4.1.2 Windowing and frequency transform

This is the same as described in subclause 6.1.1.3.4.2

6.8.1.4.1.3 Energy per band and per bin analysis

This is the same as described in subclause 6.1.1.3.4.3

6.8.1.4.1.4 Excitation type classification

This is the same as described in subclause 6.1.1.3.4.4

6.8.1.4.1.5 Inter-tone noise reduction in the excitation domain

This is the same as described in subclause 6.1.1.3.4.5

6.8.1.4.1.6 Inter-tone quantization noise estimation

This is the same as described in subclause 6.1.1.3.4.6

6.8.1.4.1.7 Increasing spectral dynamic of the excitation

This is the same as described in subclause 6.1.1.3.4.7

6.8.1.4.1.8 Per bin normalization of the spectrum energy

This is the same as described in subclause 6.1.1.3.4.8

6.8.1.4.1.9 Smoothing of the scaled energy spectrum along the frequency axis and the time axis

This is the same as described in subclause 6.1.1.3.4.9

6.8.1.4.1.10 Application of the weighting mask to the enhanced concatenated excitation spectrum

This is the same as described in subclause 6.1.1.3.4.10

6.8.1.4.1.11 Inverse frequency transform and overwriting of the current excitation

This is the same as described in subclause 6.1.1.3.4.11

6.8.1.4.2 Unvoiced and inactive post processing

When the classifier described in subclause 5.3.1 of [6] considers the synthesis as unvoiced or inactive and containing background noise, the unvoiced and inactive post processing module is used to determine a cut-off frequency where the time-domain contributions should stop. Then the content above this cut-off frequency is replaced with random noise giving a smoother rendering of the synthesis. This post processing module is used when the local attack flag (l_{af} as defined in subclause 5.3.1 [6] is set to 0 and the coding type is INACTIVE and the bitrate is below or equal to 12650 bps. It is also used at 6600 bps if the synthesis is classified as UNVOICED or VOICED_TRANSITION.

When the synthesis is considered as INACTIVE and the energy of the synthesis as defined in subclause 5.3.1 of [6] is greater than -3 dB, the LP filter coefficients that will be used to do the synthesis filtering, as described below in subclause 6.8.1.1.4.5, are smoothed as between past and current frame as follow:

$$A_q'(k) = 0.7 \cdot A_{q(t-1)}(k) + 0.3 \cdot A_q(k), \qquad for\ 0 < k < 4 \cdot (16+1) \qquad (2001)$$

where $A_{q(t-1)}$ represents the LP filter of the previous frame. At the end of the post processing $A_{q(t-1)}$ is updated using A_q'.

6.8.1.4.2.1 Frequency transform

During the frequency-domain modification phase, the excitation needs to be represented into the transform-domain. The time-to-frequency conversion is achieved with a type II DCT giving a resolution of 25Hz. The frequency representation of the time-domain CELP excitation $f_u(k)$ is given below:

$$f_u(k)=\begin{cases}\sqrt{\frac{1}{L}}\cdot\sum_{n=0}^{L-1}u(n), & k=0\\ \sqrt{\frac{2}{L}}\cdot\sum_{n=0}^{L-1}u(n)\cdot\cos\left(\frac{\pi}{L}\left(n+\frac{1}{2}\right)k\right), & 1\le k\le L-1\end{cases} \qquad (2002)$$

where $u(n)e_{td}(n)$, is the time-domain excitation, and L is the frame length and its value is 256 samples for a corresponding inner sampling frequency of 12.8 kHz.

6.8.1.4.2.2 Energy per band analysis

Before any modification to the excitation, the energy per band $E_b(i)$ is computed and kept in memory for energy adjustment after the excitation spectrum reshaping. The energy can be computed as follow :

$$E_b(i)=\sqrt{\sum_{j=C_{Bb}(i)}^{j=C_{Bb}(i)+B_b(i)}f_u(j)^2} \qquad (2003)$$

where C_{Bb} is the cumulative frequency bins per band and B_b number of bins per band defined as :

$$B_b=\{4,4,4,4,4,5,6,6,6,8,8,10,11,13,15,18,22,16,16,20,20,20,16\}$$

and

$$C_{Bb}=\begin{Bmatrix}0,8,12,16,20,25,31,37,43,51,59,69,80,93,?\\ 108,126,148,164,180,200,220,240\end{Bmatrix}$$

The low frequency bands correspond to the critical audio bands, but the frequency band above 3700 Hz are a little shorter to better match the possible spectral energy variation in those bands.

6.8.1.4.2.3 Excitation modification

6.8.1.4.2.3.1 Cut off frequency of the temporal contribution

To achieve a transparent switching between the non-modified excitation and the modified excitation for unvoiced and inactive signals, it is preferable to keep at least the lower frequencies of the temporal contribution. The frequency where the temporal contribution stop to be used, the cut-off frequency f_c, has a minimum value of 1.2 kHz. It means that the first 1.2 kHz of the decoded excitations is always kept and depending of the pitch value, this cut-off frequency can be higher. The 8th harmonic is computed from the lowest pitch of all subframes and the temporal contribution is kept up to this 8th harmonic. The estimate is performed as follow:

$$h_{8th}=\frac{(8\cdot F_s)}{\min(T(k))_{k=0}^{k=3}} \qquad (2004)$$

where $F_s=12800$ and T the decoded subframe pitch.

For all bands a verification is made to find the band in which the 8th harmonic is located by searching for the highest frequency band L_f for which the following inequality is still verified:

$$\left(h_{8^{th}} \geq L_f(i)\right) \tag{2005}$$

where the frequency band L_f is defined as :

$$L_f = \begin{Bmatrix} 175, 275, 375, 475, 600, 750, 900, 1050, 1250, 1450, 1700, 1975, \\ 2300, 2675, 3125, 3675, 4075, 4475, 4975, 5475, 5975, 6375 \end{Bmatrix}$$

The index of that band will be called $i_{8^{th}}$ and it indicates the band where the 8th harmonic is likely located. The finale cut-off frequency f_{tc} is computed as the higher frequency between the 1.2 kHz and the last frequency of the frequency band in which the 8th harmonic is located $\left(L_f\left(i_{8^{th}}\right)\right)$, using the following relation:

$$f_{tc} = \max\left(L_f\left(i_{8^{th}}\right), 1.2\,\text{kHz}\right) \tag{2006}$$

6.8.1.4.2.3.2 Normalization and noise fill

For unvoiced and inactive frames, the frequency bins below f_{tc} f_c are normalized between [0, 4] :

$$\overline{f_u}(j) = \begin{cases} \dfrac{4 \cdot f_u(j)}{\max\limits_{0 \le i < f_c}\left(\left|f_u(i)\right|\right)}, & \text{for } 0 \le j < f_{tc} \\ 0 & \text{for } f_{tc} \le j < 256 \end{cases} \tag{2007}$$

And the frequency bins above f_{tc} f_c are zeroed. Then, a simple noise fill is performed to add noise over all the frequency bins at a constant level. The function describing the noise addition is defined below as:

$$\begin{aligned} &\text{for } j = 0 : L\text{-}1 \\ &\qquad \overline{f_u}'(j) = \overline{f_u}(j) + 0.75 \cdot s_r(j) \end{aligned} \tag{2008}$$

Where s_r is a random number generator which is limited between -1 to 1 as :

$$s_r(j) = \frac{\text{float}\left(\text{short}\left[s_r(j-1) \times 31821 + 13849\right]\right)}{32768} \tag{2009}$$

6.8.1.4.2.3.3 Energy per band analysis of the modified excitation spectrum

The energy per band after the spectrum reshaping $E_b{}'$ is calculated again with exactly the same method as described in subclause 6.8.1.1.4.2.

6.8.1.4.2.3.4 Amplification of high frequencies

An amplification factor α compensates for the poor energy matching in high frequency of the LP filter at low bit rate. It is based on the voice factor V_f and computed as follow:

$$\alpha = 0.5 * \left(1 - V_f\right) \tag{2010}$$

where V_f is given by:

$$V_f = 0.34 + 0.5 \cdot \lambda + 0.16 \cdot \lambda^2 \tag{2011}$$

and λ is defined in sub-clause 6.1.1.3.2.

The amplification factor is applied linearly between 6kHz and 6.4kHz as follow:

$$\overline{f_u}''(j)=\begin{cases}\overline{f_u}'(j) & ,\ \mathit{for}\ 0\le j<240\\ \overline{f_u}'(j)\cdot\max\left(1,\ \alpha\cdot(0.067\cdot j-15.0)\right) & ,\ \mathit{for}\ 240\le j<256\end{cases} \qquad (2012)$$

6.8.1.4.2.3.5 Energy matching

The energy matching consists in adjusting the energy per band after the excitation spectrum modification to its initial value. For each bands *i*, the gain G_b to apply to all bins in the band for matching the energy of the original excitation f_u is defined as:

$$G_b(i)=\frac{E_b(i)}{E_b'(i)} \qquad (2013)$$

For a specific band ***i***, the denormalized f'_{edN} f_u' spectral excitation can be written as :

$$\begin{aligned}&\text{for } C_{Bb}(i)\le j<C_{Bb}(i)+B_b(i)\\ &\qquad f_u'(j)=G_b(i)\cdot\overline{f_u}''(j)\end{aligned} \qquad (2014)$$

where C_{Bb} and B_b are defined in subclause 6.8.1.1.4.2.

6.8.1.4.2.4 Inverse frequency transform

After the frequency domain is completed, an inverse frequency-to-time transform is performed in order to find the temporal excitation. The frequency-to-time conversion is achieved with the same type II DCT as used for the time-to-frequency conversion. The modified time-domain excitation u' is obtained as below:

$$u'(k)=\begin{cases}\sqrt{\frac{1}{L}}\cdot\sum_{n=0}^{L-1}f_u'(n), & k=0\\ \sqrt{\frac{2}{L}}\cdot\sum_{n=0}^{L-1}f_u'(n)\cdot\cos\left(\frac{\pi}{L}\left(n+\frac{1}{2}\right)k\right), & 1\le k\le L-1\end{cases} \qquad (2015)$$

where $f_u'(n)$ $f'_{edN}(n)$, is the frequency representation of the modified excitation, and L is the frame length that is equal to 256 samples.

6.8.1.5 Synthesis filtering and overwriting the current CELP synthesis

Once the excitation modification is done, the modified excitation is passed through the synthesis filter, as described in in subclause 6.1.3, to obtain a modified synthesis for the current frame. This modified synthesis is then used to overwrite the decoded synthesis.

6.8.1.6 Formant post-filter

The decoded synthesis is post-filtered as described in subclause 6.1.4.1.

6.8.1.7 Comfort noise addition

For frames exhibiting a high background noise level (background noise level >= 15), comfort noise is added for bitrates of 8.85 kbps and below. The comfort noise addition is described in subclause 6.9.1.

6.8.1.8 Bass post-filter

This is the same as described in subclause 6.1.4.2

6.8.2 Resampling

The decoded synthesis (after post-filtering and comfort noise addition) is resampled as described in subclause 6.5. Note that the bass-postfilter described in subclause 6.8.1.8 is actually realized as part of the resampling.

6.8.3 High frequency band

The high-frequency band generation for modes from 6.6 to 23.05 kbit/s is illustrated in figure 113. The high band is generated by generating an over-sampled excitation signal in DCT domain that is extended in the 6400-8000 Hz band above the 0-6400 Hz band. Note that in reality the high band is extended to a slightly wider band (6000-8000 Hz) to facilitate the addition of low and high-band, especially in the cross-over region around 6400 Hz. Tonal and ambiance components in the extended band are extracted and combined adaptively to obtain the extended excitation signal, which is then filtered in DCT domain. After inverse DCT, gains are applied in time domain (by sub-frame) and the extended excitation signal is filtered by an LP filter whose coefficients are derived from the LP filter.

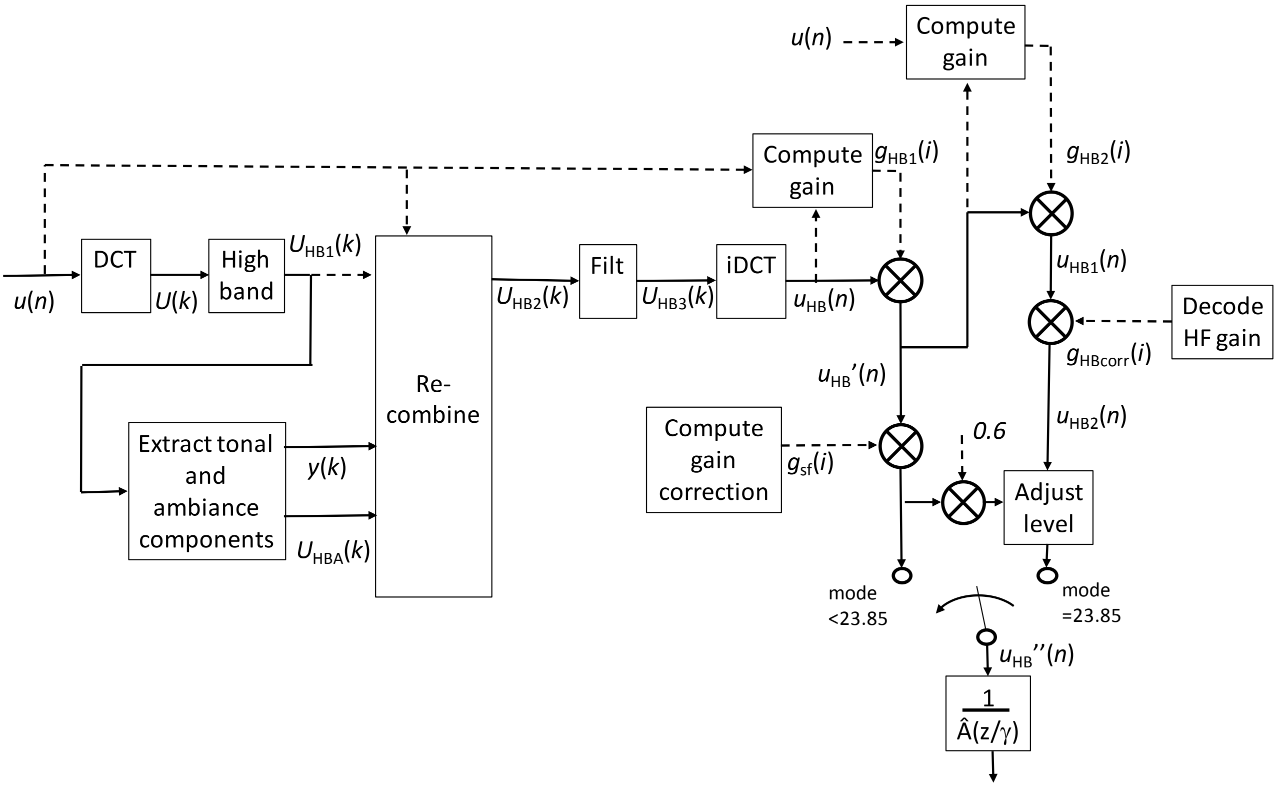


Figure 113: High-frequency band generation in AMR-WB IO modes

6.8.3.1 Preliminary estimation steps

The low-frequency band signal is extended to obtain the high frequency band signal by bandwidth extension algorithm, and the bandwidth extension algorithm includes the estimation of gains and the prediction of the excitation of the high frequency band signal.

The gains of the high frequency band signal are estimated by pitch, noise gate factor, voice factor, classification parameter and LPC.

The excitation of the high frequency band signal is adaptively predicted from the decoded low frequency band excitation signal (the sum of adaptive codebook contribution and algebraic codebook contribution) according to LSF and the bitrates.

The excitation of the high frequency band signal is modified by the gains of the high frequency band signal, and the LP synthesis is performed by filtering the modified excitation signal through the LP synthesis filter to obtain the high frequency band signal.

The parameters of estimating the gains and predicting excitation of the high frequency band signal are decoded from the bitstream of low band or calculated by the decoded low band signal.

6.8.3.1.1 Estimation of tilt, figure of merit and voice factors

1) Calculate the spectrum tilt factor of each subframe according to the decoded low frequency band as follows:

$$Til(i) = \frac{\sum_{k=L_{subfr}\times i}^{L_{subfr}\times(i+1)-2} Syn(k)\times Syn(k+1)}{\sum_{k=L_{subfr}\times i}^{L_{subfr}\times(i+1)-1} Syn(k)\times Syn(k)}, \qquad i = 0,1,2,3 \qquad (2016)$$

where L_{subfr} denotes the length of sub frame and $Syn(k)$ denotes the decoded low frequency band signal. $Til(i)$ is preserved as $Til0(i)$ for the following unvoiced flag calculation.

2) Calculate the sum of the differences between every two adjacent pitch values:

$$Sum_P^{diff} = \left|d^{[0]} - d^{[1]}\right| + \left|d^{[1]} - d^{[2]}\right| + \left|d^{[2]} - d^{[3]}\right| \qquad (2017)$$

where $d^{[i]}$ is the pitch value of each subframe.

3) If the frame counter f_{count} is greater than 100 and the FEC class of current frame is $UNVOICE\_CLAS$, set f_{count} to 0 and set the minimum noise gate $Ng_{\min}$ to -30. Otherwise, if f_{count} is greater than 200, set f_{count} to 200; if not, f_{count} is increased by 1; If the noise gate Ng is less than $Ng_{\min}$, set $Ng_{\min}$ to Ng.

4) Calculate the average voice factor as follows:

$$V_{fac}^{aver} = \frac{\sum_{i=0}^{3} V_{fac}(i)}{4} \qquad (2018)$$

where $V_{fac}(i)$ denotes the voice factor of each subframe.

5) Based on the classify parameter $fmerit$ from FEC classification, determine two parameters $fmerit_w$ and $fmerit_m$.

$$fmerit_w = \begin{cases} 0.35, & if\ \ fmerit > 0.35 \\ 0.15, & else\ if\ \ fmerit < 0.15 \\ fmerit, & otherwise \end{cases} \qquad (2019)$$

and if the FEC class of current frame is $AUDIO\_CLAS$, $fmerit_w = 0.5\times fmerit_w$.

Then $fmerit_w$ is further modified by the average voice factor V_{fac}^{aver} and smoothed as follows:

$$fmerit_w = (1+V_{fac}^{aver})\times fmerit_w \qquad (2020)$$

$$fmerit_w^{sm} = 0.9\times fmerit_w^{sm} + 0.1\times fmerit_w \qquad (2021)$$

Set $fmerit_w$ to $fmerit_w^{sm}$.

If $fmerit$ is less than 0.5, set $fmerit_m$ to 1; Otherwise, set $fmerit_m$ to $(2 - fmerit)$. Then smooth $fmerit_m$ as follows:

$$fmerit_m^{sm} = 0.5\times fmerit_m^{sm} + 0.5\times fmerit_m \qquad (2022)$$

Set $fmerit_m$ to $fmerit_m^{sm}$.

6) If the sum of the differences between every two adjacent pitch values Sum_P^{diff} is less than 10 and the spectrum tilt factor of current subframe $Til(i)$ is less than zero, reset $Til(i)$ to 0.2. Then, if $Til(i)$ is greater than 0.2, reset $Til(i)$ to 0.8; Otherwise, reset $Til(i)$ to $(1\text{-}Til(i))$. Finally, modify $Til(i)$ as follows:

$$Til(i) = (Til(i) + (30 + Ng_{\min}) \times 0.007) * fmerit_m \qquad (2023)$$

6.8.3.1.2 Estimation of sub-frame gains based on LP spectral envelopes

The signal in low-band (0-64000 Hz) is generated based on a source-filter model, where the filter is given by the synthesis filter $1/\hat{A}(z)$. Similarly, as shown in subclause 6.8.3.3, the signal in high-band (above 6400 Hz) is generated based on a source-filter model; the filter in high-band is an linear predictive (LP) filter $A_{HB}(z) = \hat{A}(z/\gamma_{HB})$ derived from the LP filter in low-band.

Since the low and high-band are combined in the final synthesis, a preliminary equalization step is performed to match the levels of the two LP filters at a given frequency. At 6400 Hz the shape of $1/\hat{A}(z)$ is already too decreasing, therefore a frequency of 6000 Hz has been chosen for this equalization frequency point.

In each sub-frame, the frequency response of the LP filter $\hat{A}(z)$ in the low-band and the LP filter $A_{HB}(z) = \hat{A}(z/\gamma_{HB})$ in the high-band are computed at the frequency of 6000 Hz:

$$R = \frac{1}{\left|\hat{A}(e^{j\theta})\right|} = \frac{1}{\left|\sum_{i=0}^{M} \hat{a}_i e^{-ji\theta}\right|}, \theta = 2\pi \frac{6000}{12800} \qquad (2024)$$

and

$$P = \frac{1}{\left|\hat{A}(e^{j\theta'}/\gamma_{HB})\right|} = \frac{1}{\left|\sum_{i=0}^{M} \hat{a}_i \, \gamma_{HB}{}^{i} e^{-ji\theta'}\right|}, \ \theta' = 2\pi \frac{6000}{16000} \qquad (2025)$$

where γ_{HB}=0.9 at 6.6 kbit/s and 0.6 at other modes (from 8.85 to 23.85 kbit/s)

These values are computed efficiently using the following pseudo-code:

```
px = py = 0
rx = ry = 0
for i=0 to 16
    px = px + Ap[i]*exp_tab_p[i]
    py = py + Ap[i]*exp_tab_p[33-i]
    rx = rx + Aq[i]*exp_tab_q[i]
    ry = ry + Aq[i]*exp_tab_q[33-i]
end for
P = 1/sqrt(px*px+py*py)
R = 1/sqrt(rx*rx+ry*ry)
```

where Aq[i]= $\hat{a}_i$ are the coefficients of $\hat{A}$(z) , Ap[i]= $\gamma_{HB}{}^{i}\,\hat{a}_i$ are the coefficients of $\hat{A}$(z/ γ_{HB}) , sqrt() corresponds to the square root operation and the tables exp_tab_p and exp_tab_q of size 34 contain the real and imaginary parts of complex exponentials at 6000 Hz:

$$\text{exp\_tab\_p[i]} = \begin{cases} \cos\left(2\pi \frac{6000}{12800} i\right) & i = 0,\Lambda ,16 \\ -\sin\left(2\pi \frac{6000}{12800}(33 - i)\right) & i = 17,\Lambda ,33 \end{cases} \qquad (2026)$$

and

$$\text{exp\_tab\_q[i]} = \begin{cases} \cos\left(2\pi\dfrac{6000}{16000}i\right) & i = 0,\Lambda\ ,16 \\ -\sin\left(2\pi\dfrac{6000}{16000}(33-i)\right) & i = 17,\Lambda\ ,33 \end{cases} \tag{2027}$$

The ratio R/P provides an estimated gain to be used in each sub-frame to align at the given frequency point (6000 Hz) the level of LP spectral envelopes in two different bands. This value is further refined to optimize overall quality.

To avoid over-estimating the sub-frame gain in high-band which could result in too high enrgy in the high hand, an additional LP filter of lower order is also computed based on the lower-band LP filter. An LP filter of order 2 is derived by truncating the filter $\hat{A}(\mathrm{z})$ decoded in low band to an order of 2 (instead of an order of 16). The stability of this truncated filter is ensured by the following steps:

- The filter is initialized as: $\hat{a}_i' = \hat{a}_i$, i=1, 2
- Reflection coefficients are computed: $k_1 = \hat{a}_1'/(1+\hat{a}_2')$, $k_2 = \hat{a}_2'$
- Filter stability and control of resonance is forced by applying the following conditions:

$$k_2 \leftarrow \begin{cases} \min(0.6, k_2) & k_2 > 0 \\ \max(-0.6, k_2) & k_2 < 0 \end{cases} \tag{2028}$$

$$k_1 \leftarrow \begin{cases} \min(0.99, k_2) & k_1 > 0 \\ \max(-0.99, k_2) & k_1 < 0 \end{cases} \tag{2029}$$

- The coefficients of the LP filter of order 2 are then given by: $\hat{a}_1' = (1+k_2)k_1$, $\hat{a}_2' = k_2$

The frequency response of the resulting LP filter of order 2 is computed as follows:

$$Q = \frac{1}{\left|\sum_{k=0}^{2} \hat{a}_k' e^{-jk\theta}\right|}, \theta = 2\pi\frac{6000}{12800} \tag{2030}$$

which can be computed efficiently using a similar pseudo-code with tables exp_tab_p and exp_tab_q It was found that, for some signals, using the value Q instead of the value R takes better into account the influence of spectral tilt in the actual signal spectrum and therefore avoids the influence of spectral peaks or valleys near the reference frequency point (6000 Hz) which could bias the value R.

The optimized gain to shape the excitation in high-band is then estimated based on R, P, Q.

Before the gain is estimated, an unvoiced flag is determined first so that the gain estimation will be different for unoiced speech and voiced speech. An unvoicing parameter is defined as,

$$P_{c\_unvoicing\_tmp} = 0.5\,(1-Til0(i))\cdot(1-P_{voicing})\cdot MIN(Til(i)/1.5-1,\quad 1) \tag{2031}$$

wherein $P_{voicing}$ is a smoothed voicing parameter of V_{fac}^{aver}. The unvoicing parameter is first smoothed by

$$P_{c\_unvoicing} = 0.5\,P_{c\_unvoicing} + 0.5\,P_{c\_unvoicing\_tmp} \tag{2032}$$

Then, it is further smoothed by,

```
if $(P_{c\_unvoicing\_sm} > P_{c\_unvoicing})$ {
    $P_{c\_unvoicing\_sm} = 0.9\,P_{c\_unvoicing\_sm} + 0.1\,P_{c\_unvoicing}$
}
else {                                                            (2033)
    $P_{c\_unvoicing\_sm} = 0.99\,P_{c\_unvoicing\_sm} + 0.01\,P_{c\_unvoicing}$
}
```

A relative difference parameter is now defined as

$$P_{c\_unvoicing\_diff} = P_{c\_unvoicing} - P_{c\_unvoicing\_sm} \tag{2034}$$

An initial unvoiced flag is decided by the following procedure,

```
if $(P_{c\_unvoicing\_diff} > 0.1)$ {
    $Unvoiced\_flag = TRUE;$
}
else if $(P_{c\_unvoicing\_diff} < 0.05)$ {
    $Unvoiced\_flag = FALSE;$                                     (2035)
}
else {
    $Unvoiced\_flag$ is not changed (previous $Unvoiced\_flag$ is kept).
}
```

A final unvoiced flag is limited to

$$Final\_Unvoiced\_flag = Unvoiced\_flag \;\; AND \;\; (R>P) \tag{2036}$$

The gain computed is performed according to the voicing of the signal:

If the sub-frame is classified as unvoiced

$$g_{sf}(i) = \min\left(5, \max\left(\min\left(R', \mathrm{Q}'\right), P\right)/P\right) \tag{2037}$$

where the smoothed value $R^{(m)}$ in the current sub-frame of index m is computed as

$$R^{(i)} = \begin{cases} 0.5R + 0.5R^{(i-1)} & R > R^{(i-1)} \\ R & otherwise \end{cases} \tag{2038}$$

and

$$R' = R^{(i)}.\min\left(1, Til(i).\left(1.6 - V_{fac}(i)\right)\right) \tag{2039}$$

and

$$Q' = Q.\max\left(1, Til(i).\left(1.6 - V_{fac}(i)\right)\right) \tag{2040}$$

Otherwise, if the sub-frame is not classified as unvoiced:

$$g_{sf}(i) = \min\left(5, \min\left(lev_1, lev_2\right)/P\right) \qquad (2041)$$

where the smoothed value $R^{(i)}$ in the current sub-frame of index i is computed as

$$R^{(i)} = (1-\alpha)R + \alpha R^{(i-1)} \qquad (2042)$$

with $\alpha = 1 - R^2$ if $R < 1$ and $R^{(m-1)} < 1$, $\alpha = 0$ otherwise, and

$$Q' = Q.\max\left(1, Til(i).\left(1.6 - V_{fac}(i)\right)\right) \qquad (2043)$$

and where

$$lev_1 = Q.\min\left(1, Til(i).\left(1.6 - V_{fac}(i)\right)\right) \qquad (2044)$$

and

$$lev_2 = \min\left(R^{(i)}, P, Q\right)\left(1 + \left|Til(i) - 1\right|.\left(1.6 - V_{fac}(i)\right)\right) \qquad (2045)$$

6.8.3.2 Generation of high-band excitation

6.8.3.2.1 DCT

The current frame of decoded excitation from the low-band, $u(n)$, $n = 0,\Lambda,255$, sampled at 12.8 kHz, is transformed in DCT domain as described in sub-clause 5.2.3.5.3.1, to obtain the spectrum, $U(k)$, $k = 0,\Lambda,255$.

6.8.3.2.2 High band generation

6.8.3.2.2.1 Adaptive start frequency bin prediction

The start frequency bin of predicting the high band excitation from the low band excitation k_{start} is adaptively determined by the line spectrum frequency (LSF) parameters. The LSF parameters are decoded from the bitstream of low frequency band. Based on the decoded LSF parameters of the low band signal, the differences between every two adjacent LSF parameters are calculated and the minimum difference is searched since the minimum difference corresponds to an energy peak of the low band spectral envelope. The start frequency bin k_{start} is determined by the position of the minimum difference, where the low band excitation is decoded from the bitstream of the low band as described in subclause 6.8.1.1.

In order to mitigate switching the start frequency bin frequently in $VOICE\_CLAS$ or $AUDIO\_CLAS$, the voicing flag F_{voice} will be determined according to the average voice factor V_{fac}^{aver} and the FEC class of current frame $class_{FEC}$:

$$F_{voice} = \begin{cases} 1 & if\ V_{fac}^{aver} > 0.4\ OR\ \left(V_{fac}^{aver} > 0.3\ \ AND\ class \geq 3\right)\ OR\ class_{FEC} = AUDIO\_CLAS \\ 0 & otherwise \end{cases} \qquad (2046)$$

The voicing flag F_{voice} is further refined to 0 if $V_{fac}^{aver} < 0.2 AND\ class_{FEC} < VOICE\_CLAS$.

Initially the start frequency bin k_{start} is 160. If the bitrate is not less than 23050, the start frequency bin $k_{start} = 160$; Otherwise, the start frequency bin k_{start} is adaptively searched as follows:

1) Calculate the LSF differences between every two adjacent LSF parameters:

$$d_{LSF}(k) = LSF(k) - LSF(k-1), \qquad k = 1,2,\Lambda,M-1 \qquad (2047)$$

where M is the order of the LP filter and $M = 16$.

2) Determine the range of search the minimum LSF difference in $d_{LSF}(k)$:

Initialize the range to $\left[2, M2\right]$, $M2 = M - 2$, if voicing flag $F_{voice} = 1$, reset $M2$:

$$M2 = \begin{cases} M-8, & if\ bitrate \le 8850 \\ M-6, & else\,if\ bitrate \le 12650 \\ M-4, & else\,if\ bitrate \le 15850 \end{cases} \tag{2048}$$

3) Search the minimum value $V_{\min}$ of the adjusted LSF difference $V_{cri}(k)$ in the range $[2, M2)$, $V_{cri}(k)$ is calculated as follows:

$$V_{cri}(k) = d_{LSF}(k) * \max(1 - LSF(k) * W, 0.001), \qquad k \in S \tag{2049}$$

and $V_{\min} = \min\limits_{k \in [2, M2)} \left(V_{cri}(k)\right)$, the position $k_{\min}$ of the minimum value $V_{\min}$ is

$$k_{\min} = \operatorname*{argmin}_{k \in [2, M2)} \left(V_{cri}(k)\right) \tag{2050}$$

where W is adjust factor of LSF parameters based on the core bitrate and the FEC class of current frame:

$$W = \begin{cases} 0.75 \cdot \left(\dfrac{bitrate}{19850}\right)^2 \Big/ 6000 & if\ class_{FEC} = AUDIO\_CLAS \\ \left(\dfrac{bitrate}{19850}\right)^2 \Big/ 6000 & otherwise \end{cases} \tag{2051}$$

4) The start frequency bin of of predicting the high band excitation from the low band excitation is calculated:

$$k_{start} = \min\left(\max\left(\left\lfloor \frac{0.5 \cdot (LSF(k_{\min}) + LSF(k_{\min} - 1)) \cdot 40}{1000} - 40 \right\rfloor, 40 \right), 160 \right) \tag{2052}$$

5) In order to decrease the distortion of the spectrum of the high band, the start frequency bin of the current frame k_{start} is reset with the start frequency bin of the previous frame $k_{start}^{[-1]}$ when the below conditions is satisfied:

- If one of the conditions $F_{voice}^{[-1]} \ne F_{voice}$, $F_{voice} = 0\ AND\, V_{\min} < V_{\min}^{[-1]}$, or $V_{\min} < 0.7 \cdot V_{\min}^{[-1]}\ AND\ V_{\min}^{[-1]} > 64$ is satisfied, $V_{\min}$ of current frame is preserved for the next frame, and

$$k_{start} = k_{start}^{[-1]}, \qquad if\ \left|k_{start} - k_{start}^{[-1]}\right| < 20\ AND\ F_{voice} = 1\ AND\ F_{voice}^{[-1]} = 1 \tag{2053}$$

- Otherwise, the start frequency bin of bandwidth extension is set to $k_{start}^{[-1]}$, and the $V_{\min}$ of current frame is preserved for the next frame if $V_{\min} < V_{\min}^{[-1]}\ AND\ F_{voice} = 1$.

The start frequency bin of predicting the high band excitation from the low band excitation is further refined if the FEC class of current frame is $AUDIO\_CLAS$:

$$k_{start} = \min(k_{start}, 120) \tag{2054}$$

If k_{start} is not an even number, k_{start} is decremented by one.

Then, obtain the high band excitation by choosing low band excitation with a given length of the bandwidth according to the start frequency bin k_{start} .

6.8.3.2.2.2 Extension of excitation spectrum

The DCT spectrum covering the 0-6400 Hz band is extended to the 0-8000 Hz band as follows:

$$U_{HB1}(k)=\begin{cases}0 & k=0,\Lambda,199\\ U(k) & k=200,\Lambda,239\\ U(k+k_{start}-240) & k=240,\Lambda,319\end{cases} \quad (2055)$$

where k_{start} is the adaptive start band as computed according to subclause 6.8.3.2.2.1. The 5000-6000 Hz band in $U_{HB1}(k)$ is copied from $U(k)$ in the same band, this allows keeping the original spectrum in this band to avoid introducing distortions when the high-band is added to the decoded low-band signal. The 6000-8000 Hz band in $U_{HB1}(k)$ is copied from $U(k)$ e.g. in the 4000-6000 Hz band when k_{start} =160.

6.8.3.2.3 Extraction of tonal and ambiance components

Tonal and ambiance components are extracted in the 6000-8000 Hz. This extraction is implemented according to the following steps:

- Computation of total energy $ener_{HB}$ in the extended low-band signal:

$$ener_{HB}=\sum_{k=240}^{319}U_{HB1}(k)^2+\varepsilon \quad (2056)$$

where ε =0.1.

- Computation of the ambiance component (in absolute value) corresponding to the average (bin-by-bin) level of the spectrum $lev(i)$ and computation of the energy $ener_{tonal}$ of dominant tonal components in high frequency:

The average level is given by the following equation:

$$lev(i)=\frac{1}{fn(i)-fb(i)+1}\sum_{j=fb(i)}^{fn(i)}\left|U_{HB1}(j+240)\right|,\ i=0..L-1 \quad (2057)$$

where L = 80. This level gives an average level in absolute value and represents a sort of spectral envelope. Note that the index $i=0..L-1$ corresponds to indices $j+240$ from 240 to 319, i.e. the 6000-8000 Hz band. In general, $fb(i)=i-7$ and $fn(i)=i+7$, however for the first and last 7 indices ($i=0,\Lambda,6$ et $i=L-7,\Lambda,\mathrm{L}-1$) the following values are used:

$fb(i)=0$ and $fn(i)=i+7$ for $i=0,\Lambda,6$

$fb(i)=i-7$ and $fn(i)=L-1$ for $i=L-7,\Lambda,\mathrm{L}-1$

- Detection and computation of the residual signal which defines tonal components:

$$\mathrm{y}(i)=\left|U_{HB1}(i+240)\right|-lev(i),\ i=0...L-1 \quad (2058)$$

Tonal components are detected using the criterion $\mathrm{y}(i)$ >0.

- Computation of the energy $ener_{tonal}$ of dominant tonal components in high frequency:

The energy of tonal components is computed as follows:

$$ener_{tonal} = \sum_{i=0...7|y(i)|>0} y(i)^2 \text{ , } i = 0..L-1 \qquad (2059)$$

6.8.3.2.4 Recombination

The extracted tonal and ambiance components are re-mixed adaptively. The combined signal is obtained using absolute values as:

$$\mathrm{y}'(i) = \begin{cases} \Gamma y(i) + \frac{1}{\Gamma} lev(i) & y(i) > 0 \\ y(i) + \frac{1}{\Gamma} lev(i) & y(i) \le 0 \end{cases} \text{ , } i = 0..L-1 \qquad (2060)$$

where the factor controlling the ambiance

$$\Gamma = \beta \frac{ener_{HB} - ener_{tonal}}{ener_{HB} - \beta ener_{tonal}} \qquad (2061)$$

and β is a multiplicative factor given by:.

$$\beta = 1 - fmerit_m \qquad (2062)$$

Tonal components, that were detected using the criterion $y(i) > 0$, are reduced by a factor Γ and the average level is amplified by $1/\Gamma$.

Signs from $U_{HB1}(k)$ are then applied as follows:

$$\mathrm{y}''(i) = \mathrm{sgn}(U_{HB1}(i+240)).y'(i) \text{ , } i = 0..L-1 \qquad (2063)$$

where

$$\mathrm{sgn}(x) = \begin{cases} 1 & x \ge 0 \\ -1 & x < 0 \end{cases} \qquad (2064)$$

The combined high-band signal $U_{HB2}(k)$ is then obtained by adjusting the energy as follows:

$$U_{HB2}(k) = fac.y''(k-240) \text{ , } k = 240,\Lambda ,319 \qquad (2065)$$

where the adjustment factor is given by:

$$fac = \gamma \sqrt{\frac{ener_{HB}}{\sum_{i=0}^{L-1} y''(i)}} \qquad (2066)$$

The factor γ is used to avoid over-estimation of energy and is given by:

$$\gamma = \max(0.3, \min(1, 1.05 \text{-} 0.95\alpha)) \qquad (2067)$$

and

$$\alpha = \sqrt{fmerit_w.(1.1 \text{-} 0.00625.\mathrm{start\_band})} \qquad (2068)$$

6.8.3.2.5 Filtering in DCT domain

The excitation is de-emphasized as follows:

$$U_{HB2}'(k)=\begin{cases}0 & k=0,\Lambda\ ,199\\ G_{deemph}(k-200)U_{HB2}(k) & k=200,\Lambda\ ,255\\ G_{deemph}(55)U_{HB2}(k) & k=256,\Lambda\ ,319\end{cases} \qquad (2069)$$

where $G_{deemph}(k)$ is the frequency responses of the filter $1/\left(1-0.68z^{-1}\right)$ over a limited frequency range. Taking into account the (odd) frequencies of the DCT, $G_{deemph}(k)$ is given by:

$$G_{deemph}(k)=\frac{1}{\left|e^{j\theta_k}-0.68\right|},\ k=0,\Lambda\ ,255 \qquad (2070)$$

where

$$\theta_k=\frac{256-80+k+\frac{1}{2}}{256},\ k=0,\Lambda\ ,255 \qquad (2071)$$

The de-emphasis is applied in two steps, for $k=200,\Lambda\ ,255$ where the response of $1/\left(1-0.68z^{-1}\right)$ is applied in the 5000-6400 Hz band, and for $k=256,\Lambda\ ,319$ corresponding to the 6400-8000 Hz band. This de-emphasis is used to bring the signal in a domain consistent with the low-band signal (in the 0-6.4 band), which is useful for the subsequent energy estimation and adjustment.

Then, the high-band is bandpass filtered in DCT domain, by splitting fixed high-pass filtering and adaptive low-pass filtering. The partial response of the low-pass filter in DCT domain is computed as follows:

$$G_{lp}(k)=1-0.999\frac{k}{N_{lp}-1},\ k=0,\Lambda\ ,255 \qquad (2072)$$

where N_{lp} =60 at 6.6 kbit/s, 40 at 8.85 kbit/s, and 20 for modes >8.85 bit/s. It defines a low-pass filter with variable cut-off frequency, depending on the mode in the current frame. Then, the band-pass filter is applied in the following form:

$$U_{HB3}(k)=\begin{cases}0 & k=0,\Lambda\ ,199\\ G_{hp}(k-200)U_{HB2}'(k) & k=200,\Lambda\ ,255\\ U_{HB2}'(k) & k=256,\Lambda\ ,319-N_{\mathrm{lp}}\\ G_{lp}(k-320-N_{\mathrm{lp}})U_{HB2}'(k) & k=320-N_{\mathrm{lp}},\Lambda\ ,319\end{cases} \qquad (2073)$$

The definition of the factor $G_{hp}(k)$, $k=0,\Lambda\ ,55$ is given in table 174.

Table 174: High-pass filter in DCT domain

k	$G_{hp}(k)$	k	$G_{hp}(k)$	k	$G_{hp}(k)$	k	$G_{hp}(k)$
0	0.001622428	14	0.114057967	28	0.403990611	42	0.776551214
1	0.004717458	15	0.128865425	29	0.430149896	43	0.800503267
2	0.008410494	16	0.144662643	30	0.456722014	44	0.823611104
3	0.012747280	17	0.161445005	31	0.483628433	45	0.845788355
4	0.017772424	18	0.179202219	32	0.510787115	46	0.866951597
5	0.023528982	19	0.197918220	33	0.538112915	47	0.887020781
6	0.030058032	20	0.217571104	34	0.565518011	48	0.905919644
7	0.037398264	21	0.238133114	35	0.592912340	49	0.923576092
8	0.045585564	22	0.259570657	36	0.620204057	50	0.939922577
9	0.054652620	23	0.281844373	37	0.647300005	51	0.954896429
10	0.064628539	24	0.304909235	38	0.674106188	52	0.968440179
11	0.075538482	25	0.328714699	39	0.700528260	53	0.980501849
12	0.087403328	26	0.353204886	40	0.726472003	54	0.991035206
13	0.100239356	27	0.378318805	41	0.751843820	55	1.000000000

6.8.3.2.6 Inverse DCT

The current frame of extended excitation in high-band, $U_{HB3}(k)$, $k = 0,\Lambda ,320$, sampled at 16 kHz, is transformed in time domain as described in subclause 5.2.3.5.13, to obtain the signal $u_{HB}(n)$, $n = 0,\Lambda ,320$.

6.8.3.2.7 Gain computation and scaling of excitation

6.8.3.2.7.1 6.6, 8.85, 12.65, 14.25, 15.85, 18.25, 19.85 or 23.05 kbit/s modes

The signal $u_{HB}(n)$, $n = 0,\Lambda ,320$ is scaled by sub-frame of 5 ms as follows:

$$u_{HB}'(n) = g_{HB1}(i)u_{HB}(n),\quad n = 80i,\Lambda ,80(i+1)-1 \tag{2074}$$

where m =0,1,2,3 is the sub-frame index and

$$g_{HB1}(i) = \sqrt{\frac{e_3(i)}{e_2(i)}} \tag{2075}$$

with

$$e_1(i) = \sum_{n=0}^{63} u(n+64i)^2 + \varepsilon$$

$$e_2(i) = \sum_{n=0}^{79} u_{HB}(n+80i)^2 + \varepsilon \tag{2076}$$

$$e_3(i) = e_1(i)\frac{\sum_{n=0}^{319} u_{HB}(n)^2 + \varepsilon}{\sum_{n=0}^{255} u(n)^2 + \varepsilon}$$

and ε = 0.01. The sub-frame gain $g_{HB1}(m)$ can be further written as:

$$g_{HB1}(i)=\sqrt{\frac{\dfrac{\sum_{n=0}^{63}u(n+64i)^2+\varepsilon}{\sum_{n=0}^{255}u(n)^2+\varepsilon}}{\dfrac{\sum_{n=0}^{79}u_{HB}(n+80i)^2+\varepsilon}{\sum_{n=0}^{319}u_{HB}(n)^2+\varepsilon}}} \qquad (2077)$$

which shows that this gain is used to have in $u_{HB}'(n)$ the same ratio of sub-frame vs frame energy than in the low-band signal $u(n)$.

The scaled extended excitation signal is then computed for $n = 80i,\cdots,80(i+1)-1$ as follows:

$$u_{HB}''(n) = g_{sf}(i).u_{HB}'(n) == g_{HB1}(i).(g_{sf}(i).u_{HB}(n)) \qquad (2078)$$

where $g_{sf}(i)$ is given in Eqs. 2037 and 2041 and $g_{sf}(i).u_{HB}(n)$ is the extended excitation signal.

6.8.3.2.7.2 23.85 kbit/s mode

In the 23.85 kbit/s mode, a high-frequency (HF) gain is transmitted at a bit rate of 0.8 kbit/s (4 bits per 5 ms sub-frame). This information is transmitted only at 23.85 kbit/s and it used in EVS AMR-WB IO to improve quality by adjusting the excitation gain.

To be able to use the HF gain information, the excitation has to be converted to a signal domain similar to AMR-WB high-band coding. To do so the energy of the excitation is adjusted in each subframe as follows:

$$u_{HB1}(n)=g_{HB2}(i)u_{HB}'(n)\,,\ n=80m,\Lambda\ ,80(i+1)-1 \qquad (2079)$$

where the sub-frame gain $g_{HB2}(i)$ is computed as:

$$g_{HB2}(i)=\sqrt{\frac{\sum_{n=0}^{63}u(n+64i)^2}{5.\sum_{n=0}^{79}u_{HB}'(n+80i)^2}} \qquad (2080)$$

The factor 5 in the the denominator is used to compensate the difference in bandwith between the signal $u(n)$ and the signal $u_{HB}'(n)$, noting that in AMR-WB the HF excitation is a white noise in the 0-8000 Hz band.

The 4-bit index in each sub-frame, $\text{index}_{\text{HF\_gain}}(m)$, transmitted at 23.85 kbit/s is demultiplexed from the bitstream and decoded as follows:

$$g_{HBcorr}(i)=2.HP\_\text{gain}(\text{index}_{\text{HF\_gain}}(i)) \qquad (2081)$$

where $HP\_\text{gain}(.)$ is the codebook used for HG gain quantization in AMR-WB, as defined in table 175.

Table 175: AMR-WB gain codebook for high band

j	$HP\_\text{gain}(j)$	j	$HP\_\text{gain}(j)$
0	0.110595703125000	8	0.342102050781250
1	0.142608642578125	9	0.372497558593750
2	0.170806884765625	10	0.408660888671875
3	0.197723388671875	11	0.453002929687500
4	0.226593017578125	12	0.511779785156250
5	0.255676269531250	13	0.599822998046875f
6	0.284545898437500	14	0.741241455078125
7	0.313232421875000	15	0.998779296875000

Then, the signal $u_{HB1}(n)$ is scaled according to this decoded HF gain as follows:

$$u_{HB2}(n) = g_{HBcorr}(i)u_{HB1}(n)\,,\;\; n = 80i,\Lambda\,,80(i+1)-1 \qquad (2082)$$

The energy of the excitation is further adjusted by sub-frame under the following conditions. A factor $fac(m)$ is computed:

$$fac(i) = \sqrt{\frac{\sum_{n=0}^{79}\left(0.6g_{sf}\,(i)u_{HB}'(n+80i)\right)^2}{\sum_{n=0}^{79}u_{HB2}(n+80i)^2}} \qquad (2083)$$

Here the term 0.6 corresponds to the average magnitude ratio between the frequency response of the de-emphasis filter $1/\left(1-0.68z^{-1}\right)$ in the 5000-6400 Hz band. Therefore, the term $\sum_{n=0}^{79}\left(0.6g_{sf}\,(i)u_{HB}'(n)\right)^2$ represents the energy of the high-band excitation that would be obtained at 23.05 kbit/s.

Based on the tilt information of the low-band signal, the scaled extended excitation signal is then computed for $n = 80i,\Lambda\,,80(i+1)-1$ as follows:

If $fac(i)$ >1 or $Til0(i)$ <0:

$$u_{HB}''(n) = u_{HB2}(n) \qquad (2084)$$

Otherwise:

$$u_{HB}''(n) = \max\left(\min\left(1,\left(1\text{-}\mathrm{Til0(i)}\right).\left(1.6\text{-}\,\mathrm{V_{fac}(m)}\right)\right),\, fac(i)\right).u_{HB2}(n) \qquad (2085)$$

6.8.3.3 LP filter for the high frequency band

The high-band LP synthesis filter $A_{HB}(z)$ is derived from the weighted low-band LP synthesis filter as follows:

$$A_{HB}(z) = \hat{A}(z/\gamma_{HB}) \qquad (2086)$$

where $\hat{A}(z)$ is the interpolated LP synthesis filter in each 5-ms sub-frame and γ_{HB}=0.9 at 6.6 kbit/s and 0.6 at other modes (from 8.85 to 23.85 kbit/s). $\hat{A}(z)$ has been computed analysing signal with the sampling rate of 12.8 kHz but it is now used for a 16 kHz signal.

6.8.3.4 High band synthesis

The scaled extended excitation signal in high-band $u_{HB}''(n)$ is filtered by $1/A_{HB}(z)$ to obtain the decoded high-band signal, which is added to synthesized low band signal to produce the synthesized output signal.

6.8.4 CNG decoding

The CNG decoding in AMR-WB-interoperable mode is described by referring to subclause 6.7.2. The CNG parameter updates in active and inactive periods is the same as described in subclasue 6.7.2.1.1. The DTX-hangover based parameter analysis is the same as described in subclause 6.7.2.1.2. The quantized logarithmic excitation energy is found from the SID frame using Δ = 2.625 and converted to linear domain using the procedure described in subclause 5.6.2.1.5. The quantized energy $\hat{\bar{E}}$ is used to obtain the smoothed quantized excitation energy E_{CN} used for CNG synthesis in the same way as described in subclause 5.6.2.1.6. The quantized ISF vector is found in the same way as described in subclause 5.7.12. The smoothed LP synthesis filter, $\hat{A}(Z)$, is then obtained in the same way as described in subclause 5.6.2.1.4 with the only difference that the ISP vector is used instead of the LSP vector. The CNG excitation signal, $e(n),\ n = 0,\ldots,L$, where L is the frame length, is generated in the same way the random excitation signal $e_r(n),\ n = 0,\ldots,L$ is generated as described in subclause 6.7.2.1.5. The comfort noise is synthesized by filtering the excitation signal, $e(n)$, through the smoothed LP synthesis filter, $\hat{A}(Z)$.

6.9 Common post-processing

6.9.1 Comfort noise addition

In this clause, we describe a post-processing technique for enhancing the quality of noisy speech coded and transmitted at bit-rates up to 13.2 kbps. At such low bit-rates, the coding of noisy speech, i.e. speech recorded with background noise, is usually not as efficient as the coding of clean speech. The decoded synthesis is usually prone to artifacts as the two different kinds of sources - the noise and the speech - cannot be efficiently coded by a coding scheme relying on a single-source model.

The comfort noise addition (CNA) consists in modelling and synthesizing the background noise at the decoder side, requiring thereby no side-information. It is achieved by estimating the level and spectral shape of the background noise at the decoder side, and by generating artificially a comfort noise in the frequency domain. In principle, the noise estimation and generation in CNA is therefore similar to the FD-CNG presented in clause 6.7.3. However, a noticeable difference is that FD-CNG is applied in DTX operations only, whereas CNA can be used in any case when coding noisy speech at bit-rates up to 13.2 kbps. The generated noise is added to the decoded audio signal and allows masking coding artifacts.

6.9.1.1 Noisy speech detection

The CNA should be triggered in noisy speech scenarios only, i.e., not in clean speech or clean music situations. To this end, a noisy speech detector is used in the decoder. It consists in estimating the long-term SNR by separately adapting long-term estimates of either the noise or the speech/music energies, depending on a VAD decision f_{VAD}.

The VAD decision is deduced directly from the information decoded from the bitstream. It is 0 if the current frame is a SID frame, a zero frame, or an IC (Inactive Coding mode, see clause 5.1.13) frame. It is 1 otherwise.

The long-term noise estimate $\bar{N}_{\mathrm{NSD}}$ and long-term speech/music estimate $\bar{S}_{\mathrm{NSD}}$ are initialized with -20 dB and +25 dB, respectively. When $f_{\mathrm{VAD}} = 0$, the long-term noise energy is updated on a frame-by-frame basis as follows:

$$\bar{N}_{\mathrm{NSD}} = 0.995\,\bar{N}_{\mathrm{NSD}} + 0.005 \cdot 10\log_{10}\left(\sum_{i=0}^{L_{\mathrm{shaping}}-1} N_{\mathrm{FD-CNG}}^{[\mathrm{shaping}]}(i)\left(j_{\max}^{[\mathrm{shaping}]}(i) - j_{\min}^{[\mathrm{shaping}]}(i) + 1\right)\right), \quad (2086)$$

where $N_{\mathrm{FD-CNG}}^{[\mathrm{shaping}]}(i)$ refers to the noise energy spectrum estimated in the decoder to apply FD-CNG, L_{shaping} is the number of spectral partitions, and $j_{\max}^{[\mathrm{shaping}]}(i) - j_{\min}^{[\mathrm{shaping}]}(i) + 1$ corresponds to the size of each partition (see clause 6.7.3.2.2). Otherwise, i.e. if $f_{\mathrm{VAD}} = 1$, the long-term speech/music energy is updated on a frame-by-frame basis as follows:

$$\bar{S}_{\mathrm{NSD}} = 0.995\bar{S}_{\mathrm{NSD}} + 0.005 \times 10\log_{10}\left(\frac{2}{L_{\mathrm{celp}}}\sum_{n=0}^{L_{\mathrm{celp}}-1}\left(s_{\mathrm{celp}}(n)\right)^2\right), \tag{2087}$$

where L_{celp} denotes the frame size in samples and $s_{\mathrm{celp}}(n)$ is the output frame of the core decoder at the CELP sampling rate. Furthermore, the long-term noise estimate $\bar{N}_{\mathrm{NSD}}$ is lower limited by $\bar{S}_{\mathrm{NSD}} - 45$ for each frame.

The flag for noisy speech detection is set to 1 if the SNR is smaller than 28dB, i.e.

$$f_{\mathrm{NSD}} = \begin{cases} 1 & \text{if } \bar{S}_{\mathrm{NSD}} - \bar{N}_{\mathrm{NSD}} < 28 \\ 0 & \text{otherwise} \end{cases}. \tag{2088}$$

6.9.1.2 Noise estimation for CNA

To be able to produce an artificial noise resembling the actual input background noise in terms of spectro-temporal characteristics, the CNA needs an estimate of the noise spectrum in each FFT bin.

6.9.1.2.1 CNA noise estimation in DTX-on mode when FD-CNG is triggered

In DTX-on mode and provided that FD-CNG is triggered, the FD-CNG noise levels $N_{\mathrm{FD-CNG}}^{[\mathrm{CNG}]}(j),\ j = 0,\ldots,j_{\max}^{[\mathrm{SID}]}(L_{\mathrm{SID}}^{[\mathrm{FFT}]} - 1) - 1$ can be directly used. As described in clause 6.7.3, they are obtained by capturing the fine spectral structure of the background noise present during active phases, while updating only the spectral envelop of the noise during inactive parts with the help of the SID information.

6.9.1.2.2 CNA noise estimation in DTX-on mode when LP-CNG is triggered

To enable tracking of the noise spectrum when LP-CNG is triggered in DTX-on mode, the FD-CNG noise estimation algorithm (see clause 6.7.3.2.2) is applied at the output of the LP-CNG during inactive frames, yielding noise estimates $N_{\mathrm{FD-CNG}}^{[\mathrm{shaping}]}(i)$ in each spectral partition $i = 0,\ldots,L_{\mathrm{shaping}}-1$. Following the technique described in clause 6.7.3.2.3.1., the parameters $N_{\mathrm{FD-CNG}}^{[\mathrm{shaping}]}(i)$ are then interpolated to yield the full-resolution FFT power spectrum $N_{\mathrm{FD-CNG}}^{[\mathrm{shaping,FR}]}(j)$, which overwrites the current FD-CNG levels, i.e. $N_{\mathrm{FD-CNG}}^{[\mathrm{CNG}]}(j) = N_{\mathrm{FD-CNG}}^{[\mathrm{shaping,FR}]}(j)$.

6.9.1.2.3 CNA noise estimation in DTX-off mode

In DTX-off mode, the noise estimates $N_{\mathrm{FD-CNG}}^{[\mathrm{shaping}]}(i)$ are obtained by applying the FD-CNG noise estimation algorithm at the output of the core decoder when $f_{\mathrm{VAD}} = 0$ only, i.e. during speech pauses. As in the previous clause, the interpolation techniques described in clause 6.7.3.2.3.1 is then used to obtain a full-resolution FFT power spectrum $N_{\mathrm{FD-CNG}}^{[\mathrm{shaping,FR}]}(j)$, which overwrites the current FD-CNG levels, i.e. $N_{\mathrm{FD-CNG}}^{[\mathrm{CNG}]}(j) = N_{\mathrm{FD-CNG}}^{[\mathrm{shaping,FR}]}(j)$.

6.9.1.3 Noise generation in the FFT domain and addition in the time domain

In CNA and when the current frame is not a MDCT-based TCX frame, a random noise is generated in the FFT domain, separately for the real and imaginary parts. This is the same approach as in the FD-CNG (see clause 6.7.3.3.2). The noise is then added to the decoder output after performing an inverse FFT transform of the random noise using the overlap-add method.

The level of added comfort noise should be limited to preserve intelligibility and quality. The comfort noise is hence scaled to reach a pre-determined target noise level. Typically, the decoded audio signal exhibits a higher SNR than the original input signal, especially at low bit-rates where the coding artifacts are the most severe. This attenuation of the noise level in speech coding is coming from the source model paradigm which expects to have speech as input. Otherwise, the source model coding is not entirely appropriate and won’t be able to reproduce the whole energy of no-speech components. Hence, the amount of additional comfort noise is adjusted to roughly compensate for the noise attenuation inherently introduced by the coding process. The assumed amount of noise attenuation g_{CNA} is chosen depending on the bandwidth and the bit-rate, as shown in the tables below.

Table 176: Assumed noise attenuation level for EVS primary modes

Bandwidth	**NB**				**WB**				**SWB**	
Bit-rates [kbps]	< 8	8	9.6	13.2	< 8	8	9.6	13.2	≤ 9.6	13.2
g_{CNA} **[dB]**	-3.5	-3	-2.5	-2	-3	-2.5	-1.5	-2.5	-2	-1

Table 177: Assumed noise attenuation level for EVS primary modes for AMR-WB IO modes

Bandwidth	**AMR-WB IO**	
Bit-rates [kbps]	6.60	8.85
g_{CNA} **[dB]**	-4	-3

The energy $N_{\text{CNA}}(j)$ of the random noise is adjusted for each FFT bin j as

$$N_{\text{CNA}}(j) = \lambda_{\text{NSD}} \cdot \left(10^{-g_{\text{CNA}}/10} - 1\right) \cdot N_{\text{FD-CNG}}^{[\text{CNG}]}(j)\,, \tag{2089}$$

where

$$\lambda_{\text{NSD}} = 0.99\lambda_{\text{NSD}} + 0.01 f_{\text{NSD}} \tag{2090}$$

can be interpreted as the likelihood of being in a noisy speech situation. It is used as a soft decision to reject clean speech or music situations where the noisy speech detection flag f_{NSD} becomes zero (see clause 6.9.1.1).

6.9.1.4 Noise generation and addition in the MDCT domain

If the current frame is an MDCT based TCX frame, the comfort noise addition is performed directly in the MDCT domain. The random noise adjustment for each MDCT bin j is derived from the FFT-based comfort noise adjustment:

$$N_{CNA,MDCT}(j) = N_{CNA}(j) \cdot \sqrt{160}\,. \tag{2091}$$

The adjusted random noise is subsequently added to the MDCT bins as last steps before doing the inverse transformation to time-domain:

$$X(j) = X(j) + N_{CNA,MDCT}(j)\,. \tag{2092}$$

6.9.2 Long term prediction processing

For the TCX coding mode and bitrates up to 48kbps, LTP post filtering is applied to the output signal, using the LTP parameters transmitted in the bitstream.

6.9.2.1 Decoding LTP parameters

If LTP is active, integer pitch lag d_{LTP}, fractional pitch lag f_{LTP} and gain g_{LTP} are decoded from the transmitted indices $I_{LTP,lag}$ and $I_{LTP,gain}$:

$$d_{LTP} = \begin{cases} p_{\min} + \left\lfloor \frac{I_{LTP,lag}}{p_{res}} \right\rfloor & , \ if\ I_{LTP,lag} < \left(p_{fr2} - p_{\min}\right)p_{res} \\ p_{fr2} + \left\lfloor \frac{I_{LTP,lag} - \left(p_{fr2} - p_{\min}\right)p_{res}}{p_{res}/2} \right\rfloor & , \ if\ 0 \le \left(I_{LTP,lag} - \left(p_{fr2} - p_{\min}\right)p_{res}\right) < \left(p_{fr1} - p_{fr2}\right)\frac{p_{res}}{2} \\ p_{fr1} + I_{LTP,lag} - \left(p_{fr2} - p_{\min}\right)p_{res} - \left(p_{fr1} - p_{fr2}\right)\frac{p_{res}}{2} & , \ if\ I_{LTP,lag} \ge \left(p_{fr1} - p_{fr2}\right)\frac{p_{res}}{2} + \left(p_{fr2} - p_{\min}\right)p_{res} \end{cases} \tag{2093}$$

$$f_{LTP} = \begin{cases} I_{LTP,lag} - (d_{LTP} - p_{\min})p_{res} & , \ if\ I_{LTP,lag} < \left(p_{fr2} - p_{\min}\right)p_{res} \\ 2\left(I_{LTP,lag} - \left(p_{fr2} - p_{\min}\right)p_{res} - \left(d_{LTP} - p_{fr2}\right)\frac{p_{res}}{2}\right) & , \ if\ 0 \le \left(I_{LTP,lag} - \left(p_{fr2} - p_{\min}\right)p_{res}\right) < \left(p_{fr1} - p_{fr2}\right)\frac{p_{res}}{2} \\ 0 & , \ if\ I_{LTP,lag} \ge \left(p_{fr1} - p_{fr2}\right)\frac{p_{res}}{2} + \left(p_{fr2} - p_{\min}\right)p_{res} \end{cases}$$

$$g_{LTP} = 0.15625\left(I_{LTP,gain} + 1\right) \tag{2094}$$

If LTP is not active, LTP parameters are set as follows:

$$\begin{aligned} & if\ \left(LTP_{on} = 0\right) then \\ & \quad d_{LTP} = p_{\max} \\ & \quad f_{LTP} = 0 \\ & \quad g_{LTP} = 0 \end{aligned}$$

On encoder side the pitch lag is computed on the LTP sampling rate, therefore it has to be converted to the output sampling rate first:

$$\begin{gathered} t = \left\lfloor \frac{\left(d_{LTP}\, p_{res} + f_{LTP}\right)L_{out} + L_{out}/2}{N_{LTP}} \right\rfloor \\ d'_{LTP} = \left\lfloor \frac{t}{p_{res}} \right\rfloor \quad , \quad f'_{LTP} = t \bmod p_{res} \end{gathered} \tag{2095}$$

For 48kbps bitrate the LTP gain is reduced as follows:

$$g_{LTP} = \begin{cases} 0.32 g_{LTP} & , \ if\ \left(bitrate = 48000\right) \wedge \left(N_{LTP} = 320\right) \\ 0.4 g_{LTP} & , \ if\ \left(bitrate = 48000\right) \wedge \left(N_{LTP} = 512\right) \\ 0.64 g_{LTP} & , \ else \end{cases} \tag{2096}$$

6.9.2.2 LTP post filtering

For long-term prediction with fractional pitch lags polyphase FIR interpolation filters are used to interpolate between past synthesis samples. For each combination of LTP sampling rate and output sampling rate a different set of filter coefficients is used. The index i_{filt} of the interpolation filter to use is determined according to the following table:

Table 178: LTP index i_{filt} of the interpolation filter

	$sr_{LTP}=12800$	$sr_{LTP}=16000$	$sr_{LTP}=25600$
$sr_{out}=8000$	0	4	8
$sr_{out}=16000$	1	5	9
$sr_{out}=32000$	2	6	10
$sr_{out}=48000$	3	7	11

The predicted signal s_{pred} is computed by filtering the past synthesis signal with the selected FIR filter. The filtered range of the past synthesis signal is determined by the integer part of the pitch lag d'_{LTP}. The polyphase index of the filter is determined by the fractional part of the pitch lag f'_{LTP}.

The filtered signal s_{filt} is computed by low-pass filtering the current synthesis signal with polyphase index 0 of the selected interpolation filter, so that its frequency response matches the one of the predicted signal.

Both s_{pred} and s_{filt} are multiplied with the LTP gain g_{LTP}. The filtered signal is then subtracted from the synthesis signal , the predicted signal is added to it.

If both LTP gain and pitch lag are the same as in the previous frame, the full frame can be processed the same way:

$$\begin{aligned}
s_{pred}(n) &= \sum_{i=0}^{l_{filt}-1}\left(h_{LTP}^{(i_{filt},f'_{LTP})}(p_{res}i)s_{LTP}(n-d'_{LTP}+i)+h_{LTP}^{(i_{filt},p_{res}-f'_{LTP})}(p_{res}i)s_{LTP}(n-d'_{LTP}-1-i)\right)\\
s_{filt}(n) &= \sum_{i=0}^{l_{filt}-1}\left(h_{LTP}^{(i_{filt},0)}(p_{res}i)s(n+i)+h_{LTP}^{(i_{filt},p_{res})}(p_{res}i)s(n-1-i)\right)\\
s_{LTP}(n) &= s(n)+g_{LTP}\left(s_{pred}(n)-s_{filt}(n)\right) \qquad , \quad n=0..L_{out}-1
\end{aligned} \tag{2097}$$

However, if gain and/or pitch lag have changed compared to the previous frame, a 5ms transition is used to smooth the parameter change. If no delay compensation is needed, the transition starts at the beginning of the frame. If a delay of D_{LTP} needs to be compensated, the transition starts at offset D_{LTP} from the beginning of the frame. In that case the signal part before the transition is processed using the LTP parameters of the previous frame:

$$\begin{aligned}
s_{pred}(n) &= \sum_{i=0}^{l_{filt}-1}\left(h_{LTP}^{\left(i_{filt}^{(prev)},f'^{(prev)}_{LTP}\right)}(p_{res}i)s_{LTP}\left(n-d'^{(prev)}_{LTP}+i\right)+h_{LTP}^{\left(i_{filt}^{(prev)},p_{res}-f'^{(prev)}_{LTP}\right)}(p_{res}i)s_{LTP}(n-d'_{LTP}-1-i)\right)\\
s_{filt}(n) &= \sum_{i=0}^{l_{filt}-1}\left(h_{LTP}^{\left(i_{filt}^{(prev)},0\right)}(p_{res}i)s(n+i)+h_{LTP}^{\left(i_{filt}^{(prev)},p_{res}\right)}(p_{res}i)s(n-1-i)\right)\\
s_{LTP}(n) &= s(n)+g_{LTP}^{(prev)}\left(s_{pred}(n)-s_{filt}(n)\right) \qquad , \quad n=0..D_{LTP}-1
\end{aligned} \tag{2098}$$

If the LTP gain of the previous frame is zero (i.e. LTP was inactive in the previous frame), a linear fade-in is used for the gain in the transition region:

$$s_{LTP}(n)=s(n)+\frac{4(n-D_{LTP})}{L_{out}}g_{LTP}\left(s_{pred}(n)-s_{filt}(n)\right) \qquad , \quad n=D_{LTP}..D_{LTP}+\frac{L_{out}}{4}-1 \tag{2099}$$

If the LTP gain of the current frame is zero (LTP is inactive, but was active in the previous frame), a linear fade-out is used for the gain in the transition region, using the LTP parameters of the previous frame:

$$
\begin{aligned}
s_{pred}(n) &= \sum_{i=0}^{l_{filt}-1} \left(h_{LTP}^{\left(i_{filt}^{(prev)}, f_{LTP}'^{(prev)}\right)}(p_{res} i) s_{LTP}\left(n - d_{LTP}'^{(prev)} + i\right) + h_{LTP}^{\left(i_{filt}^{(prev)}, p_{res} - f_{LTP}'^{(prev)}\right)}(p_{res} i) s_{LTP}\left(n - d_{LTP}' - 1 - i\right) \right) \\
s_{filt}(n) &= \sum_{i=0}^{l_{filt}-1} \left(h_{LTP}^{\left(i_{filt}^{(prev)}, 0\right)}(p_{res} i) s(n+i) + h_{LTP}^{\left(i_{filt}^{(prev)}, p_{res}\right)}(p_{res} i) s(n-1-i) \right) \qquad (2100) \\
s_{LTP}(n) &= s(n) + \frac{L_{out} - 4(n - D_{LTP})}{L_{out}} g_{LTP}^{(prev)} \left(s_{pred}(n) - s_{filt}(n) \right) \quad , \quad n = D_{LTP} .. D_{LTP} + \frac{L_{out}}{4} - 1
\end{aligned}
$$

If LTP is active in previous and current frame and LTP parameters have changed, a zero input response z is used to smooth the transition.

The LPC coefficients for zero input LP filtering are computed from the past 20ms LTP output before the beginning of the transition, using autocorrelation and Levinson-Durbin algorithm as described in 5.1.9.

$$
\begin{aligned}
z_{pred}(n) &= \sum_{i=0}^{l_{filt}-1} \left(h_{LTP}^{\left(i_{filt}, f_{LTP}'\right)}(p_{res} i) s_{LTP}\left(n - d_{LTP}' + i\right) + h_{LTP}^{\left(i_{filt}, p_{res} - f_{LTP}'\right)}(p_{res} i) s_{LTP}\left(n - d_{LTP}' - 1 - i\right) \right) \\
z_{filt}(n) &= \sum_{i=0}^{l_{filt}-1} \left(h_{LTP}^{\left(i_{filt}, 0\right)}(p_{res} i) s(n+i) + h_{LTP}^{\left(i_{filt}, p_{res}\right)}(p_{res} i) s(n-1-i) \right) \qquad (2101) \\
z(n) &= \left(s(n) - g_{LTP} s_{filt}(n) \right) - \left(s_{LTP}(n) - g_{LTP} s_{pred}(n) \right) \quad , \quad n = D_{LTP} - m .. D_{LTP} - 1
\end{aligned}
$$

The zero input response is then computed by LP synthesis filtering with zero input, and applying a linear fade-out to the second half of the transition region:

$$
z(n) = -\min\left(\frac{2L_{out} - 8(n - D_{LTP})}{L_{out}}, 1 \right) \sum_{j=1}^{m} a(j) z(n-j) \quad , \quad n = D_{LTP} .. D_{LTP} + \frac{L_{out}}{4} - 1 \qquad (2102)
$$

Finally the output signal in the transition region is computed by LTP filtering using the current frame parameters and subtracting the zero input response:

$$
s_{LTP}(n) = s(n) + g_{LTP} \left(s_{pred}(n) - s_{filt}(n) \right) - z(n) \quad , \quad n = D_{LTP} .. D_{LTP} + \frac{L_{out}}{4} - 1 \qquad (2103)
$$

6.9.3 Complex low delay filter bank synthesis

The analysis stage of the CLDFB is described in sub-clause 5.1.2.1. The synthesis stage transforms the time-frequency matrix of the complex coefficients $X_{CR}(t,k)$ and $X_{CI}(t,k)$ to the time domain. The combination of analysis and synthesis is used for sample rate conversions. Also adaptive sample rate conversions are handled by the CLDFB, including sample rate changes in the signal flow.

The sample rate of the reconstructed output signal $s_{Crec}(n)$ depends on the number of bands L_{Cs} used for the synthesis stage, i.e. $sr_{Cs} = L_{Cs} \cdot 800Hz$. In case, $L_{Cs} > L_{Ca}$ (number of bands in analysis stage), the coefficients $> L_{Ca}$ are initialized to zero before synthesizing.

For the synthesis operation, a demodulated vector $z_t(n)$ is computed for each time step t of the sub-bands.

$$
z_t(n) = \frac{1}{2} \cdot \frac{1}{L_{Cs}} \left[\sum_{k=0}^{k=L_{Cs}-1} X_{CR}(t,k) \cos\left[\frac{\pi}{L_{Cs}} (n + n_0)\left(k + \frac{1}{2} \right) \right] + \sum_{k=0}^{k=L_{Cs}-1} X_{CI}(t,k) \sin\left[\frac{\pi}{L_{Cs}} (n + n_0)\left(k + \frac{1}{2} \right) \right] \right] \qquad (2104)
$$

$$
for\ n = 0..10L_{Cs} - 1
$$

where n_0 is identical to the one defined for the analysis operation (see 5.1.2.1). The vector is then windowed by the filter bank prototype to prepare the overlap-add operation

$$z_{wt}(n) = w_c(n) \cdot z_t(n) \quad for\ n = 0..10L_{Cs} - 1 \tag{2105}$$

Then the recent ten windowed vectors are combined in an overlap-add operation to reconstruct the signal from the CLDFB coefficients.

$$s_{Crec}(n) = \sum_{t=0}^{-9} z_{wt}(n + t \cdot L_{Cs}) \quad for\ n = 0..L_{Cs} - 1 \tag{2106}$$

6.9.4 High pass filtering

At the final stage, the signal is high pass filtered to generate the final output signal. The high pass operation is identical to the one used in the pre-processing of the EVS encoder as described in 5.1.1.

7 Description of the transmitted parameter indices

7.1 Bit allocation for the default option

The allocation of the bits for various operating modes in the EVS encoder is shown for each bitrate in the following tables. Note that the most significant bit (MSB) of each codec parameter is always sent first. In the tables below, the abbreviation CT is used to denote the coder type and the abbreviation BW is used to denote the bandwidth.

7.1.1 Bit allocation at VBR 5.9, 7.2 – 9.6 kbps

The EVS codec encodes NB and WB content at 7.2 and 8.0 kbps with CELP core or HQ-MDCT core. No extension layer is used at these bitrates. The EVS codec encodes NB and WB content at 9.6 kbps with CELP core or TCX core. To encode WB signals at 9.6 kbps, the CELP core uses TBE extension layer and the TCX core uses IGF extension layer. Similarly to encode SWB signals at 9.6 kbps, the CELP core uses TBE extension layer and the TCX core uses IGF extension layer.

VBR mode uses 4 different active frame types with different bit rates to achieve the average bit rate of 5.9 kbps. The 4 different frame rates are 2.8 kbps PPP frame, 2.8 kbps NELP frame, and 7.2 kbps and 8 kbps CELP frames. The CT bits are allocated as 1 bit to differentiate active 2.8 kbps (PPP or NELP) frames from any other 2.8 kbps frames (such as SID frame with payload header) and the remaining 2 bits are used to represent NB PPP, WB PPP, NB NELP and WB NELP frames.

Table 179: Bit allocation at 7.2 – 9.6 kbps and 2.8 kbps PPP/NELP

Description	**2.8 PPP**	**2.8 NELP**	**7.2**	**8.0**	**9.6**		
core	**CELP**	**CELP**	**CELP HQ-MDCT**	**CELP HQ-MDCT**	**CELP**		**TCX**
ext. layer	**NO**	**NO**	**NO**	**NO**	**SWB TBE**	**WB TBE**	**IGF**
Number of bits per frame	56	56	144	160	192		
BW			4		2		
CT	3	3			3		
core bits	53	53	140	156	171	181	187
WB/SWB ext. layer bits					16	6	

Note that the BW and CT parameters are combined together to form a single index at 7.2 and 8.0 kbps. This index conveys the information whether CELP core or HQ-MDCT core is used. At 9.6 kbps, the information about using the CELP core or the TCX core is encoded as a part of the CT parameter.

7.1.2 Bit allocation at 13.2 kbps

The EVS codec encodes NB, WB and SWB content at 13.2 kbps with CELP core, HQ-MDCT core, or TCX core. For WB signals, the CELP core uses TBE or FD extension layer. For SWB signals, the CELP core uses TBE or FD extension layer, and the TCX core uses IGF extension layer.

Table 180: Bit allocation at 13.2 kbps

Description	13.2						
core	CELP	HQ-MDCT	TCX	CELP			TCX
ext. layer	NO	NO	NO	WB TBE	WB FD	SWB TBE SWB FD	IGF
Number of bits per frame	264						
BW, CT, RF	5						
TCX/HQ-MDCT core flag		1	1				1
TCX CT			2				2
TD/FD ext. layer flag				1	1	1	
core bits	259	258	256	238	252	227	256
WB/SWB ext. layer bits				20	6	31	

Note that the BW, CT, and RF parameters are combined together to form a single index. This index also conveys the information whether LP-based core or MDCT-based core (TCX or HQ-MDCT) is used. The decision between the HQ-MDCT core and the TCX core is encoded with one extra bit called MDCT core flag. At this bitrate, the TCX coder type is encoded with 2 extra bits (TCX CT).

7.1.3 Bit allocation at 16.4 and 24.4 kbps

The EVS codec encodes NB, WB, SWB and FB content at 16.4 and 24.4 kbps with CELP core, HQ-MDCT core or TCX core. For SWB and FB signals, the CELP core uses TBE extension layer and the TCX core uses IGF extension layer.

Table 181: Bit allocation at 16.4 kbps

Description	16.4					
core	**CELP**	**TCX**	**HQ-MDCT**	**CELP**	**TCX**	**CELP**
ext. layer	**NO**	**NO**	**NO**	**SWB TBE**	**IGF**	**FB TBE**
Number of bits per frame	328					
BW	2					
Reserved flag	1					
CT	3	4	2	3	4	3
core bits	322	321	323	286	321	287
SWB ext. layer bits				33		31
FB ext. layer bits						4
Padding bits				3		

Table 182: Bit allocation at 24.4 kbps

Description	24.4					
core	**CELP**	**TCX**	**HQ-MDCT**	**CELP**	**TCX**	**CELP**
ext. layer	**NO**	**NO**	**NO**	**SWB TBE**	**IGF**	**FB TBE**
Number of bits per frame	488					
BW	2					
Reserved flag	1					
CELP/MDCT core flag	1					
TCX/HQ-MDCT core flag		1	1		1	
CELP->HQ core switching flag			1-2			
CT	2	2		2	2	2
core bits	482	481	481-2	422	481	423
SWB ext. layer bits				57		55
FB ext. layer bits						4
Padding bits				3		

The information about using the CELP core or the MDCT-based core (HQ-MDCT or TCX) is transmitted as a 1-bit CELP/MDCT core flag. In the case of MDCT-based core, the next bit decides whether HQ-MDCT core or TCX core is used. In the case of TCX, the remaining 2 bits are used to represent the TCX coder type (TCX CT). In the case of HQ-MDCT core, the next one or two bits signal whether the previous frame was encoded with the CELP core or not. The second bit is used to signal its internal sampling rate (12.8 or 16 kHz) only when the previous frame was encoded with the CELP core.

7.1.4 Bit allocation at 32 kbps

The EVS codec encodes WB, SWB and FB content at 32 kbps with CELP core, HQ-MDCT core, or TCX core. For SWB and FB signals, the CELP core uses TBE or FD extension layer and the TCX core uses IGF extension layer.

Table 183: Bit allocation at 32 kbps

Description	32					
core	**CELP**	**HQ-MDCT**	**TCX**	**CELP**	**TCX**	**CELP**
ext. layer	**NO**	**NO**	**NO**	**SWB TBE/FD**	**IGF**	**FB TBE/FD**
Number of bits per frame	640					
CELP/MDCT core flag	1					
CELP->HQ core switching flag		1-2				
TCX/HQ-MDCT core flag		1	1		1	
BW	4	2	2	4	2	4
CT			2		2	
TBE/FD ext. layer flag				1		
core bits	634	632-3	632	602	633	576
SWB ext. layer bits				55/31		55/31
FB ext. layer bits						4

The information about using the CELP core or the MDCT-based core (HQ-MDCT or TCX) is transmitted as a 1-bit CELP/MDCT core flag. If CELP core is selected, the BW and CT parameters are combined together to form a single index. In the case of MDCT-based core, the next bit decides whether HQ-MDCT core is used or the TCX core is used. In the case of TCX, the remaining 2 bits are used to represent the TCX coder type (TCX CT). In the case of HQ-MDCT core, the next one or two bits signal whether the previous frame was encoded with the CELP core or not. The second bit is used to signal its internal sampling rate (12.8 or 16 kHz) only when the previous frame was encoded with the CELP core. Finally, 1 bit is used to distinguish between TBE and FD extension layer in the case of CELP core.

7.1.5 Bit allocation at 48, 64, 96 and 128 kbps

The EVS codec encodes WB, SWB and FB content at 48 kbps with TCX core only. For SWB and FB signals, the TCX core uses IGF extension layer. At 64 kbps, the EVS codec encodes WB, SWB and FB content with CELP core or HQ-MDCT core. For SWB and FB signals, the CELP core uses FD extension layer.

Table 184: Bit allocation at 48, 64, 96 and 128 kbps

Description	48		64			96		128	
core	TCX	TCX	CELP	HQ-MDCT	CELP	TCX	TCX	TCX	TCX
ext. layer	NO	IGF	NO	NO	SWB FD FB FD	NO	IGF	NO	IGF
Number of bits per frame	960		1280			1920		2560	
CELP/MDCT core flag			1						
CELP->HQ core switching flag				1-2					
TCX/HQ-MDCT core flag				1					
BW	2		4	2	4	2		2	
CT									
Reserved flag	1					1		1	
TCX CT	3					3		3	
core bits	954	954	1275	1274-5	954	1914	1914	2554	2554
ext. layer bits					326				

At 64 kbps, the information about using the CELP core or the HQ-MDCT core is transmitted as a 1-bit CELP/MDCT core flag. If CELP core is selected, the BW and CT parameters are combined together to form a single index. In the case of HQ-MDCT core, the next one or two bits signal whether the previous frame was encoded with the CELP core or not. The second bit is used to signal its internal sampling rate (12.8 or 16 kHz) only when the previous frame was encoded with the CELP core.

7.2 Bit allocation for SID frames in the DTX operation

The SID payload consists of 48 bits independent of the bandwidth, bit rate and mode. The EVS codec supports three types of SID frames, one for the FD-CNG and two for the LP-CNG scheme.

Table 185: Bit allocation of FD-CNG SID frame

Description	FD-CNG
Number of bits per frame	48
CNG type flag	1
Bandwidth indicator	2
CELP sample rate	1
Global gain	7
Spectral band energy	37

The CNG type flag determines the usage of FD-CNG or LP-CNG. The bandwidth indicator indicates NB, WB, SWB or FB. The CELP sample rate can be 12.8 kHz or 16 kHz. The remaining bits are used for the spectral envelope information.

Table 186: Bit allocation of LP-CNG SID frame

Description	WB SID	SWB SID
Number of bits per frame	48	48
CNG type flag	1	1
Bandwidth indicator	1	1
Core sampling rate indicator	1	1
Hangover frame counter	3	3
LSF bits	29	29
Low-band energy bits	7	7
Low-band excitation spectral envelope bits	6	N/A
High-band energy bits	N/A	4
Unused bits	N/A	2

The CNG type flag determines if the SID belongs to FD-CNG or LP-CNG. The bandwidth indicator indicates whether the SID is a WB or a SWB SID. The core sampling rate indicator indicates whether the core is running at 12.8 kHz or 16 kHz sampling rate. The hangover frame counter indicates the number of hangover frames preceding the SID. The low-band excitation spectral envelope bits are only applicable to WB SID. The high-band energy bits are only applicable to SWB SID.

7.3 Bit allocation for the AMR-WB-interoperable option

The AMR-WB-interoperable option has the same bit allocation as AMR-WB. For more details see clause 7 of [9].

7.4 Bit Allocation for the Channel-Aware Mode

The EVS codec encodes WB and SWB content at 13.2 kbps channel aware mode with CELP core or TCX core for the primary frame as well as the partial redundant frame (RF). For both WB and SWB signals, the CELP core uses TBE extension layer and the TCX core uses IGF extension layer.

The [BW, CT, and RF] information is packed in 5 bits. When RF flag is set to zero, the channel aware mode at 13.2 kbps will be a bit exact implementation of the EVS 13.2 kbps mode described in subclause 7.1.2. An ACELP partial RF information can be transmitted along with an ACELP or a TCX primary copy. Similarly, a TCX partial RF information can be transmitted along with an ACELP or a TCX primary copy. The RF frame offset information (i.e., offset = 2 or 3, or 5, or 7) at which the partial copy is transmitted with the primary frame is included in the bit stream. Similarly, the RF frame type with 3 bits that signals (RF_NO_DATA, RF_TCXFD, RF_TCXTD1, RF_TCXTD2, RF_ALLPRED, RF_NOPRED, RF_GENPRED, and RF_NELP) is included in the bit stream. Depending on the RF frame type, the distribution of number of bits used for primary copy and partial RF information varies. The last three bits in the bit stream contains the RF frame type information. The two bits before the RF frame type information contains the RF offset data. The signalling [BW, CT, and RF] is carried in the first 5 bits in the bit stream for ease of parsing by the JBM.

Table 187: Bit allocation at 13.2 kbps channel aware mode

Description	13.2 channel aware		
core	CELP		TCX
ext. layer	WB TBE	SWB TBE	IGF
Number of bits per frame	264		
BW, CT, RF	5		
core bits (primary)	183-248	171-236	189-254
WB/SWB ext. layer bits (primary)	6	18	
Core bits (partial RF)	0-60	0-60	0-65
WB/SWB ext. layer bits (partial RF)	0-5	0-5	
RF offset	2		
RF frame type	3		

Annex A (normative): RTP Payload Format and SDP Parameters

A.0 General

This Annex describes a generic RTP payload format and SDP parameters for the EVS codec. The EVS RTP payload format consists of the RTP header, the EVS payload header, and the EVS payload data.

The byte order used in this specification is the network byte order, i.e., the most significant byte is transmitted first. The bit order is most significant bit first. This practice is presented in all figures as having the most significant bit located left-most on each line and indicated with the lowest number.

A.1 RTP Header Usage

The format of the RTP header is specified in RFC 3550 [30]. This EVS RTP payload format uses the fields of the RTP header in a manner consistent with the usages in RFC 3550 [30].

The timestamp clock frequency for the EVS codec is 16 kHz, regardless of the audio bandwidth. The duration of one speech frame-block is 20 ms for both EVS Primary and EVS AMR-WB IO modes. Thus, the timestamp is increased by 320 for each consecutive frame-block.

The RTP header marker bit (M) shall be set to 1, if the first frame-block carried in the RTP packet contains a speech frame, which is the first in a talkspurt. For all other RTP packets the marker bit shall be set to zero (M=0).

A.2 EVS RTP Payload Format

The EVS RTP Payload Format includes a Compact format and a Header-Full format, which are used depending on the required functionalities within a session and whether only a single frame is transmitted. These two formats can be switched during a session by the media sender, if the EVS RTP Payload Format is not restricted to use only the Header-Full format, as described in Annex A.3 and TS 26.114 [13].

In addition to the EVS RTP Payload Format, RFC 4867 [15] format shall also be supported for the EVS AMR-WB IO modes to provide the backward interoperability with legacy AMR-WB terminals.

The media sender is the entity encoding the audio signal frames and sending the RTP packets including the encoded frames. The media receiver is the entity receiving the RTP packets and decoding the audio signal frames from the encoded frames.

The media receiver may send Codec Mode Requests (CMRs) in the Compact format (in the 3-bit CMR) or in the Header-Full format (in the CMR byte) to the media sender for adapting the bit rate, the audio bandwidth or the operational mode (EVS primary or EVS AMR-WB IO).

A.2.1 EVS codec Compact Format

In the Compact format, the RTP payload consists of exactly one coded frame for the EVS Primary mode, and one coded frame and one 3-bit CMR field for the EVS AMR-WB IO mode. The Compact format uses protected payload sizes that uniquely identify EVS codec modes (EVS Primary or EVS AMR-WB IO mode) and bit-rates. The protected payload sizes are used for determining the bit-rate of a received coded frame at the receiver.

Table A.1 shows the protected payload sizes and the corresponding bit-rates to be used for Compact RTP payload format.

Table A.1: Protected payload sizes

Mode	Payload Size (bits)	Bitrate (kbps)
EVS Primary	48	2.4 (EVS Primary SID)
Special case (see clause A.2.1.3)	56	2.8
EVS AMR-WB IO	136	6.6
EVS Primary	144	7.2
EVS Primary	160	8
EVS AMR-WB IO	184	8.85
EVS Primary	192	9.6
EVS AMR-WB IO	256	12.65
EVS Primary	264	13.2
EVS AMR-WB IO	288	14.25
EVS AMR-WB IO	320	15.85
EVS Primary	328	16.4
EVS AMR-WB IO	368	18.25
EVS AMR-WB IO	400	19.85
EVS AMR-WB IO	464	23.05
EVS AMR-WB IO	480	23.85
EVS Primary	488	24.4
EVS Primary	640	32
EVS Primary	960	48
EVS Primary	1280	64
EVS Primary	1920	96
EVS Primary	2560	128

A.2.1.1 Compact format for EVS Primary mode

In the Compact format for EVS Primary mode, the RTP payload consists of exactly one coded frame. Hence, the coded frame follows the RTP header without any additional EVS RTP payload header.

The payload represents a speech frame of 20 ms encoded with the EVS codec bit-rate identified by the payload size. The bits are in the same order as produced by the EVS encoder, where the first bit is placed left-most immediately following the RTP header.

A.2.1.2 Compact format for EVS AMR-WB IO mode (except SID)

In the Compact format for EVS AMR-WB IO mode, except SID, the RTP payload consists of one 3-bit CMR field, one coded frame, and zero-padding bits if necessary.

A.2.1.2.1 Representation of Codec Mode Request (CMR) in Compact format for EVS AMR-WB IO mode

The 3-bit CMR field carries the codec mode request information to signal to the media sender the requested AMR-WB [37] or EVS AMR-WB IO codec mode to be applied for encoding. The signalling of AMR-WB and EVS AMR-WB IO with the 3-bit CMR field is defined as shown in Table A.2. The 3-bit CMR field in Compact format for EVS AMR-WB IO mode comprises a 3-bit element [c(0), c(1), c(2)] for signalling codec mode requests for the following EVS AMR-WB IO or AMR-WB codec modes.

Table A.2: 3-bit signalling element and EVS AMR-WB IO/AMR-WB CMR

C(0)	C(1)	C(2)	Requested Mode
0	0	0	6.6
0	0	1	8.85
0	1	0	12.65
0	1	1	15.85
1	0	0	18.25
1	0	1	23.05
1	1	0	23.85
1	1	1	none

Due to the 3-bit limitation, there is not enough signalling space for all EVS AMR-WB IO codec modes. Consequently, CMRs in Compact format for EVS AMR-WB IO are limited to include the most frequently used set of EVS AMR-WB IO /AMR-WB modes as shown in Table A.2. CMRs for EVS AMR-WB IO / AMR-WB modes 14.25 and 19.85 are not supported in Compact format for EVS AMR-WB IO. In case a request needs to be transmitted for either mode, it should be re-mapped to the next lower mode (12.65 and 18.25, respectively). Alternatively, the CMR byte in the Header-Full format may be used to transmit CMRs to 14.25 and 19.85 modes.In case of restrictions in the allowed codec modes by the mode-set MIME parameter, the 3-bit CMR for a not supported mode may be re-mapped to the next lower mode in this mode-set.

Codec mode requests for EVS primary modes shall be made using the CMR byte in the Header-Full format.

The codec mode request indicated in the 3-bit-CMR shall comply with the media type parameters (the allowed bit-rates for EVS AMR-WB IO or AMR-WB) that are negotiated for the session. When a 3-bit-CMR is received, requesting a bit-rate that does not comply with the negotiated media parameters, it shall be ignored.

A 3-bit CMR indicates the highest EVS AMR-WB IO codec mode that the media receiver (CMR sender) wants to receive. When receiving a 3-bit CMR (except value "none") the media sender shall use the EVS AMR-WB IO operation mode. The media sender should use the EVS AMR-WB IO codec mode (bit rate) requested in the received 3-bit CMR and shall not use a higher codec mode (higher bit rate). The media sender may use a lower EVS AMR-WB IO codec mode within the negotiated mode-set.

CMR code-point "none" is specified as equivalent to no CMR-value being sent. The receiver of "none" shall ignore it.

NOTE: The meaning of "none" and "NO_REQ" (see A.2.2.1.1 below) for EVS is not equivalent to code-point "CMR=15" for AMR and AMR-WB, as specified according to TS 26.114 and RFC 4867 with its errata. MGWs in the path, repacking between the RTP format according to RFC 4867 and the RTP format according to the present document, translate between these code-points.

A.2.1.2.2 Payload structure of Compact EVS AMR-WB IO mode frame

In order to minimize the need for bit re-shuffling in media gateways in case of payload format conversion to or from AMR-WB bandwidth-efficient format according to [15], the speech data bits are inserted after CMR, starting with bit d(1). Speech data bit d(0) is appended after the last speech data bit.

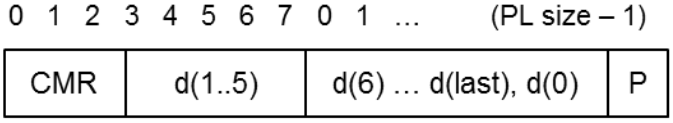


Figure A.1. Payload structure of Compact EVS AMR-WB IO.

The speech data payload represents a speech frame of 20 ms encoded with EVS AMR-WB IO bit-rate (mode) identified by the payload size. The order and numbering notation of the bits are as specified for Interface Format 1 (IF1) in Annex B of [36] for AMR-WB. The bits of the speech frames are arranged in the order of decreasing sensitivity, giving a re-ordered bit sequence $\{d(0),d(1),...,d(K-1)\}$.

If a total of three CMR bits and coded frame bits is not a multiple of 8, zero-padding bits are added so that the total becomes a multiple of 8. One zero-padding bit is required for EVS AMR-WB IO mode 6.6 and four zero-padding bits are required for EVS AMR-WB IO mode 8.85. In other mode no padding bits are inserted. With the exception of SID frames, the EVS AMR-WB IO Compact payload follows the RTP header without any additional EVS RTP payload header.

Note that no Compact frame format EVS AMR-WB IO SID frames is defined. For such frames the Header-Full format with CMR byte shall be used (see clause A.2.1.3).

NOTE: The Q bit defined in RFC 4867 [15] is not present in the Compact payload structure of EVS AMR-WB IO. Therefore it shall be ensured that the speech payload is not damaged. In case of a conversion of RFC 4867 formatted packets to Compact payload format, damaged frames (indicated by the Q bit) shall be discarded and not converted.

A.2.1.3 Special case for 56 bit payload size (EVS Primary or EVS AMR-WB IO SID)

The Compact format for EVS Primary 2.8 kbps frames (56 bits) has the same payload size (56 bits) as the Header-Full format for EVS AMR-WB IO SID frames with CMR byte.
Hence, two types of frames can be carried in the 56 bit payload case:

- EVS Primary 2.8 kbps frame in Compact format.
- EVS AMR-WB IO SID frame in Header-Full format (see clause A.2.2) with one CMR byte.
- The payload structure and bit ordering of EVS Primary 2.8 kbps frame in Compact format is defined in Figure A.2.

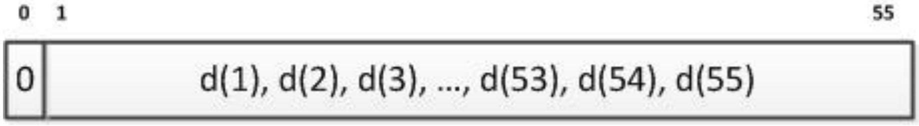


Figure A.2. Payload structure for EVS Primary 2.8 kbps (56-bit) payload

The resulting ambiguity between EVS Primary 2.8 kbps and EVS AMR-WB IO SID frames is resolved through the most significant bit (MSB) of the first byte of the payload. By definition, the first data bit d(0) of the EVS Primary 2.8 kbps is always set to ‘0’. Therefore, if the MSB of the first byte of the payload is set to ‘0’ (see Figure A.2), then the payload is an EVS Primary 2.8 kbps frame in Compact format. Otherwise it is an EVS AMR-WB IO SID frame in Header-Full format with one CMR byte. The structure of EVS AMR-WB IO SID frame with Header-Full format is described in clause A.2.2.

A.2.2 EVS codec Header-Full format

In the Header-Full format, the payload consists of one or more coded frame(s) with EVS RTP payload header(s). There are two types of EVS RTP payload header: Table of Content (ToC) byte and Codec Mode Request (CMR) byte. The detailed header structure is described in clause A.2.2.1.

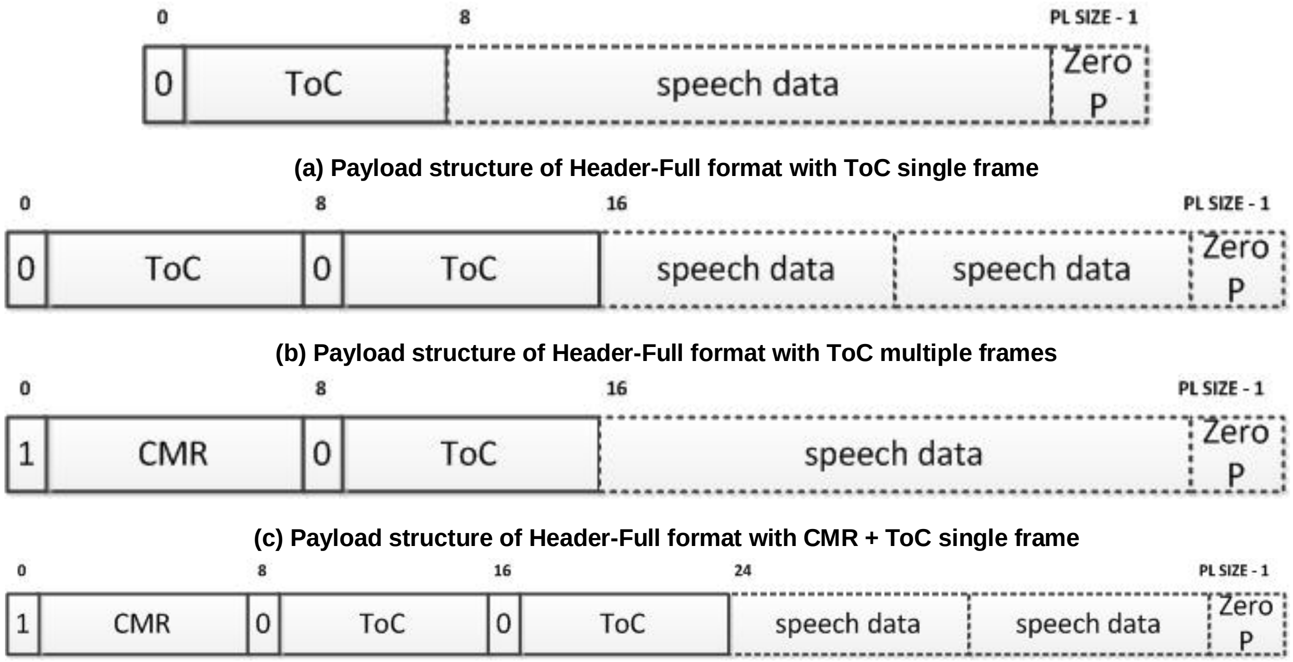
A.2.2.1 EVS RTP payload structure

The complete payload of Header-Full EVS frames comprises an optional CMR byte, followed by one or several ToC bytes, followed by speech data bits and possible zero-padding bits. Padding bits shall be discarded by the receiver. The purpose of padding is two-fold:

- In the case of EVS AMR-WB IO frames, payload data may need to be octet-aligned using zero-padding bits at the end of the payload. Note that EVS Primary frames are by definition octet-aligned (see clause A.2.2.1.4.1).
- When required, zero-padding bits are also used to increase the total payload size by byte increments such that conflicts with any of the protected sizes reserved for the Compact format are avoided (see clause A.2.2.1.4.2).

CMR and ToC bytes use MSB as Header Type identification bit (H) in order to identify the type of EVS RTP payload header. If the H bit is set to 0, the corresponding byte is a ToC byte, and if set to 1, the corresponding byte is a CMR byte. A CMR byte, if present, shall be located before ToC byte(s).

Figure A.3 shows the general structure of Header-Full payload format.



(a) Payload structure of Header-Full format with ToC single frame

(b) Payload structure of Header-Full format with ToC multiple frames

(c) Payload structure of Header-Full format with CMR + ToC single frame

(d) Payload structure of Header-Full format with CMR + ToC multiple frames

Figure A.3 Payload structure of Header-Full format

NOTE: The zero padding at the end of packet, indicated in Figure A.3 as "Zero P", does not represent the octet-alignment for AMR-WB IO data described in clause A.2.2.1.4.1, but it represents the zero-padding for size collision avoidance described in clause A.2.2.1.4.2.

A.2.2.1.1 CMR byte

The Codec Mode Request (CMR) byte structure is shown in Figure A.4. This CMR byte shall be present for EVS AMR-WB IO speech and SID frames in Header-Full format. For EVS Primary mode, the CMR byte is only used when a CMR needs to be transmitted or if required by session negotiation. The request indicated in the CMR byte shall comply with the media type parameters (e.g. allowed bit-rates or audio bandwidths) that are negotiated in the session.

NOTE 1: There is no SDP MIME signalling parameter defined that can be used to disallow all CMRs with T-bits "001". However, the mode-set MIME parameter can be used to restrain the allowed EVS AMR-WB IO codec modes. If this mode-set parameter is not included in the media type parameters, then all 9 modes of the EVS AMR-WB IO codes modes are allowed.

The media receiver in the MTSI terminal shall be prepared to receive any speech frames within the negotiated media type parameter set as well as SID frames, irrespective of the CMR it sends or receives.

NOTE 2: The media receiver can receive such frames for various reasons. For instance, after a handover to AMR-WB, a MGW can send speech frames with an EVS AMR-WB IO codec mode even if it receives CMR byte of EVS Primary mode (T-bits not "001").

The bit-rate in the CMR byte of EVS Primary mode (T-bits not "001") indicates the highest bit-rate that the media receiver (CMR sender) wants to receive. The media sender should use the bit-rate requested in the received CMR and shall not use a higher bit-rate. The media sender may use a lower bit-rate than the requested bit-rate within the set of negotiated bit-rates.

If a single audio bandwidth is negotiated for EVS Primary mode, the CMR shall indicate this bandwidth in its T-bits, unless the mode of operation is switched by a received CMR from EVS Primary to EVS AMR-WB IO or is kept in EVS AMR-WB IO operation mode.

If a range of audio bandwidths is negotiated for EVS Primary mode, then the audio bandwidth in the CMR byte of EVS Primary mode indicates the highest audio bandwidth that the media receiver wants to receive. The media sender should use the audio bandwidth requested in the received CMR.

A CMR with T-bits "001" (i.e. a CMR for the EVS AMR-WB IO mode of operation) indicates the highest EVS AMR-WB IO codec mode that the media receiver wants to receive. When receiving a CMR with T-bits "001", the media sender shall use the EVS AMR-WB IO mode of operation. The media sender should use the EVS AMR-WB IO codec mode (bit rate) requested in the received CMR and shall not use a higher codec mode (higher bit rate). The media sender may use a lower EVS AMR-WB IO codec mode within the negotiated mode-set.

When a CMR is received, requesting a bit-rate and/or audio bandwidth that does not comply with the negotiated media parameters, it shall be ignored.

The request in the received CMR is valid until a new request is received.

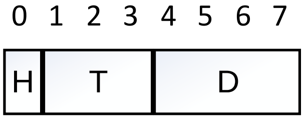


Figure A.4. CMR byte

H (1 bit): Header Type identification bit. For the CMR byte this bit is always set to 1.

T (3 bits): These bits indicate the Type of Request in order to distinguish EVS AMR-WB IO and EVS Primary bandwidths.

D (4 bits): These bits indicate the requested bit rate (in cases the T-bits are "000", "001", "010", "011" and "100") or the EVS Channel Aware offset and level (in cases the T-bits are "101" and "110") of the codec mode request.

The possible values of the CMR byte and corresponding CMRs are defined in Table A.3.

Table A.3: Structure of the CMR byte

Code		Definition		Code		Definition	
T	D			T	D		
000	0000	NB	5.9 (VBR)	010	0000	WB	5.9 (VBR)
	0001	NB	7.2		0001	WB	7.2
	0010	NB	8.0		0010	WB	8
	0011	NB	9.6		0011	WB	9.6
	0100	NB	13.2		0100	WB	13.2
	0101	NB	16.4		0101	WB	16.4
	0110	NB	24.4		0110	WB	24.4
	0111		Not used		0111	WB	32
	1000		Not used		1000	WB	48
	1001		Not used		1001	WB	64
	1010		Not used		1010	WB	96
	1011		Not used		1011	WB	128
	1100		Not used		1100		Not used
	1101		Not used		1101		Not used
	1110		Not used		1110		Not used
	1111		Not used		1111		Not used
001	0000	IO	6.6	011	0000		Not used
	0001	IO	8.85		0001		Not used
	0010	IO	12.65		0010		Not used
	0011	IO	14.25		0011	SWB	9.6
	0100	IO	15.85		0100	SWB	13.2
	0101	IO	18.25		0101	SWB	16.4
	0110	IO	19.85		0110	SWB	24.4
	0111	IO	23.05		0111	SWB	32
	1000	IO	23.85		1000	SWB	48
	1001		Not used		1001	SWB	64
	1010		Not used		1010	SWB	96
	1011		Not used		1011	SWB	128
	1100		Not used		1100		Not used
	1101		Not used		1101		Not used
	1110		Not used		1110		Not used
	1111		Not used		1111		Not used

Table A.3: Structure of the CMR byte (continued)

Code		Definition		Code		Definition	
T	D			T	D		
100	0000		Not used	110	0000	SWB	13.2 CA-L-O2
	0001		Not used		0001	SWB	13.2 CA-L-O3
	0010		Not used		0010	SWB	13.2 CA-L-O5
	0011		Not used		0011	SWB	13.2 CA-L-O7
	0100		Not used		0100	SWB	13.2 CA-H-O2
	0101	FB	16.4		0101	SWB	13.2 CA-H-O3
	0110	FB	24.4		0110	SWB	13.2 CA-H-O5
	0111	FB	32		0111	SWB	13.2 CA-H-O7
	1000	FB	48		1000		Not used
	1001	FB	64		1001		Not used
	1010	FB	96		1010		Not used
	1011	FB	128		1011		Not used
	1100		Not used		1100		Not used
	1101		Not used		1101		Not used
	1110		Not used		1110		Not used
	1111		Not used		1111		Not used
101	0000	WB	13.2 CA-L-O2	111	0000	Reserved	
	0001	WB	13.2 CA-L-O3		0001	Reserved	
	0010	WB	13.2 CA-L-O5		0010	Reserved	
	0011	WB	13.2 CA-L-O7		0011	Reserved	
	0100	WB	13.2 CA-H-O2		0100	Reserved	
	0101	WB	13.2 CA-H-O3		0101	Reserved	
	0110	WB	13.2 CA-H-O5		0110	Reserved	
	0111	WB	13.2 CA-H-O7		0111	Reserved	
	1000		Not used		1000	Reserved	
	1001		Not used		1001	Reserved	
	1010		Not used		1010	Reserved	
	1011		Not used		1011	Reserved	
	1100		Not used		1100	Reserved	
	1101		Not used		1101	Reserved	
	1110		Not used		1110	Reserved	
	1111		Not used		1111	NO_REQ	

CMR code-point "NO_REQ" is specified as equivalent to no CMR-value being sent. The receiver of "NO_REQ" shall ignore it.

NOTE: The meaning of "NO_REQ" and “none” (see A.2.1.2.1 above) for EVS is not equivalent to code-point "CMR=15" for AMR and AMR-WB, as specified according to TS 26.114 and RFC 4867with its errata. MGWs in the path, repacking between the RTP format according to RFC 4867 and the RTP format according to the present document, translate between these code-points.

A.2.2.1.2 ToC byte

The Table of Content (ToC) byte structure is shown in Figure A.5.

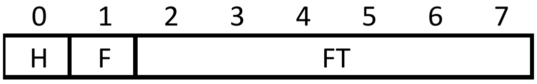


Figure A.5. ToC byte

H (1 bit): Header Type identification bit. For the ToC byte this bit is always set to 0.

F (1 bit): If set to 1, the bit indicates that the corresponding frame is followed by another speech frame in this payload, implying that another ToC byte follows this entry. If set to 0, the bit indicates that this frame is the last frame in this payload and no further header entry follows this entry.

FT (6 bits): Frame type index. These bits indicate whether the EVS Primary or EVS AMR-WB IO mode, or comfort noise (SID) mode of the corresponding frame is carried in this payload. FT is further divided into 3 fields: EVS mode (1 bit), Unused/Q bit (1 bit) depending on the value of EVS mode bit, and EVS bit-rate (4 bits). The value of FT is defined in Tables A.4 and A.5.

Table A.4: Frame Type index when EVS mode bit = 0

EVS mode bit (1 bit)	Unused (1 bit)	EVS bit rate	Indicated EVS mode and bit rate
0	0	0000	Primary 2.8 kbps
0	0	0001	Primary 7.2 kbps
0	0	0010	Primary 8.0 kbps
0	0	0011	Primary 9.6 kbps
0	0	0100	Primary 13.2 kbps
0	0	0101	Primary 16.4 kbps
0	0	0110	Primary 24.4 kbps
0	0	0111	Primary 32.0 kbps
0	0	1000	Primary 48.0 kbps
0	0	1001	Primary 64.0 kbps
0	0	1010	Primary 96.0 kbps
0	0	1011	Primary 128.0 kbps
0	0	1100	Primary 2.4kbps SID
0	0	1101	For future use
0	0	1110	SPEECH_LOST
0	0	1111	NO_DATA

Table A.5: Frame Type index when EVS mode bit = 1

EVS mode bit (1 bit)	AMR-WB Q bit (1 bit)	EVS bit rate (4 bits)	Indicated EVS mode and codec mode
1	Q	0000	AMR-WB IO 6.6 kbps
1	Q	0001	AMR-WB IO 8.85 kbps
1	Q	0010	AMR-WB IO 12.65 kbps
1	Q	0011	AMR-WB IO 14.25 kbps
1	Q	0100	AMR-WB IO 15.85 kbps
1	Q	0101	AMR-WB IO 18.25 kbps
1	Q	0110	AMR-WB IO 19.85 kbps
1	Q	0111	AMR-WB IO 23.05 kbps
1	Q	1000	AMR-WB IO 23.85 kbps
1	Q	1001	AMR-WB IO 2.0 kbps SID
1	Q	1010	For future use
1	Q	1011	For future use
1	Q	1100	For future use
1	Q	1101	For future use
1	Q	1110	SPEECH_LOST
1	Q	1111	NO_DATA

NOTE: The 4-bit EVS bit-rate index and the mapping to EVS AMR-WB IO codec mode in Table A.4 are the same as used for the Frame Type of AMR-WB. See Table 1a [36]. The Q bit for EVS AMR-WB IO has the same definition as in [15]. If Q bit is set to 0, this indicates that the corresponding frame is severely damaged. The receiver should handle such a severly damaged frame properly by applying bad frame processing according to [6].

Packets containing only NO_DATA frames should not be transmitted in any payload format configuration, except for situations, when CMR needs to be sent immediately. Frame-blocks containing only NO_DATA frames at the end of the packet should not be transmitted in any payload format configuration. In addition, frame blocks containing only NO_DATA frames in the beginning of the packet should not be included in the payload.

For sessions with multiple mono-channels, see clause A.2.5.

A.2.2.1.3 Speech Data

In Header-Full format, the RTP payload comprises, apart from headers and possible padding, one or several coded frames, the Speech Data.

In case the frame is coded EVS Primary mode data, the bits are in the same order as produced by the EVS encoder, where the first bit is placed left-most immediately following the EVS RTP payload header (CMR byte if present, and ToC bytes).

In case the frame is coded EVS AMR-WB IO mode data, the Speech Data field is constructed as described in RFC 4867 [15] for octet-aligned Mode, sub-clause 4.4.3. In accordance with this, in case multiple frames are included in the payload, the last octet of each frame shall be padded with zero bits at the end if some bits in the octet are not used. The padding bits shall be ignored on reception.

In case the frame is coded EVS AMR-WB IO SID data, the payload structure and bit-ordering are defined in Figure A.6. The bits *d*(0) to *d*(39) are as defined in TS 26.201 [36], sub-clause 4.2.3.

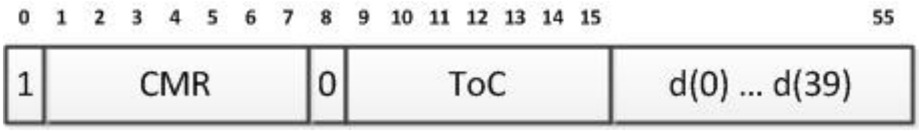


Figure A.6. Payload structure for EVS AMR-WB IO SID (56 bit) payload

The EVS AMR-WB IO SID frame payload is identified by MSB of the first byte of the payload set to ‘1’.

A.2.2.1.4 Zero padding

A.2.2.1.4.1 Zero padding for octet alignment of speech data (EVS AMR-WB IO)

In EVS AMR-WB IO mode, the payload length is always made an integral number of octets by padding with zero bits if necessary (see clause A.2.2.1.3).

Note that, by definition, EVS Primary speech data is octet-aligned.

A.2.2.1.4.2 Zero padding for size collision avoidance

When “hf-only=0” or “hf-only” is not present, the RTP payload formatting function of the sender shall control the size of Header-Full RTP payload so that the Header-Full format RTP payload size does not collide with any of the protected Compact format RTP payload sizes listed in Table A.1, except for the special case of the 56-bit payload. If a Header-Full format RTP payload size collides with one of the protected Compact format RTP payload sizes, the RTP payload formatting function of the sender shall append an appropriate number of zero-padding bytes to the end of the payload such that payload sizes do not collide.

The Header-Full format representing an EVS AMR-WB IO SID frame (with one CMR byte and one ToC byte) is allowed to have the same 56 bits as EVS Primary 2.8 kbps in Compact format. In this special case, no padding bits shall be appended to the EVS AMR-WB IO SID frame.

A.2.2.1.4.3 Additional zero padding

If additional padding is required to bring the payload length to a larger multiple of octets or for some other purposes, then the P bit in the RTP header may be set and padding bits are appended as specified in [30].

A.2.3 Header-Full/Compact format handling

There are two format handling modes: Default mode and Header-Full-only mode.

A.2.3.1 Default format handling

When “hf-only=0” is present or when the “hf-only” attribute is not present, the Compact format shall be used in the following cases:

- A single mono EVS Primary mode frame is carried in an RTP packet without sending CMR.
- A single mono EVS AMR-WB IO mode frame with 3-bit CMR is carried in an RTP packet.

Otherwise, the Header-Full format with size collision avoidance shall be used.

The only exception in this default format handling is as follows: the Header-Full format may be used to transmit a single EVS AMR-WB IO frame to request 14.25 or 19.85 kbps in EVS AMR-WB IO mode as these two bit-rates cannot be indicated with the 3-bit CMR defined for Compact format.

A.2.3.2 Header-Full-only format handling

When “hf-only=1” is present, only the Header-Full format shall be used during the session. In other words, the Compact format shall not be used. The size collision avoidance shall not be performed by the RTP payload formatting function of the sender. The RTP payload decoding function of the receiver shall use ToC byte(s) to obtain the mode (i.e., EVS Primary or EVS AMR-WB IO) and the bit-rate regardless of the RTP payload size.

A.2.4 AMR-WB backward compatible EVS AMR-WB IO mode format

In order to provide backward interoperability with AMR-WB, the payload format in [15] shall also be supported for EVS AMR-WB IO mode. This payload format shall be used to communicate with a terminal not supporting EVS but supporting AMR-WB.

A.2.5 Sessions with multiple mono channels

The Header-Full EVS payload format supports transmission of multiple mono channels in the same way as described in the AMR-WB payload format [15].

A.2.5.1 Encoding of multiple mono channels

The speech encoders for different channels are not synchronized, which means that they may use different codec modes and may result in different VAD decisions depending on the content in each channel.

A.2.5.2 RTP header usage

The RTP time stamp is derived from the media time of the first frame of the first channel in the packet, even if that frame is a NO_DATA frame.

If a frame in the packet is an onset frame, then the Marker bit in the RTP header is set to ‘1’. However, since the encoders are not synchronized, they may use different VAD decisions for different channels. Hence, it is not sufficient to only use the Marker bit to detect onset frames, and to for example reset the jitter buffers in the receiver. The receiver needs to monitor the content of the channels, e.g., the Frame Type identifier, to find the transition from DTX to active speech for each individual channel.

A.2.5.3 Construction of the RTP payload

The ToC bytes of all frames from a frame-block are placed in consecutive order as defined in Section 4.1 [38]. Therefore, with N channels and K speech frame-blocks in a packet, there shall be N*K ToC bytes in the EVS RTP payload header, and the first N ToC bytes will be from the first frame-block, the second N ToC bytes will be from the second frame-block, and so on.

The payload shall include frames from all channels for each media time that is included. If a frame is not available for a channel, e.g., when the encoder for that channel is currently in DTX mode, then a NO_DATA frame shall be included instead. Since the payload always contains two or more frames, the Header-Full payload format shall be used.

The payload may contain a CMR byte according to the same rules as defined for single-channel session. When a CMR is received, it is applied equally to all channels. It may still happen that different channels are encoded in different modes, especially if independent encoders are used.

A.2.6 Storage Format

The storage format is used for storing EVS Primary or EVS AMR-WB IO speech frames in a file or as an email attachment. Multiple channel content is supported.

For EVS AMR-WB IO, the storage format of [15] can be used.

For EVS, the storage format has the following structure:

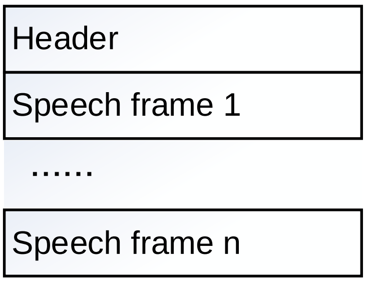


Figure A.7. Storage format for EVS

There is another storage format that is suitable for applications with more advanced demands on the storage format, like random access or synchronization with video. This format is the 3GPP-specified ISO-based multimedia file format specified in [40]. Its media type is specified in [41].

A.2.6.1 Header

The header consists of a magic number followed by a 32-bit channel description field, giving the header the following structure:

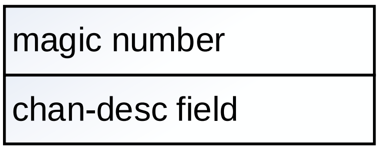


Figure A.8. Header for EVS

The magic number shall consist of the ASCII character string:

"#!EVS_MC1.0\n" or (0x23214556535f4d43312e30)

The version number in the magic number string refers to the version of the file format.

The 32-bit channel description field is defined as a 32-bit number (unsigned integer, MSB first). This number indicates the number of audio channels contained in this storage file starting from 1 for mono to N for a multi-mono signal with N channels.

A.2.6.2 Speech Frames

After the header, speech frame-blocks consecutive in time are stored in the file. Each frame-block contains a number of octet-aligned speech frames equal to the number of channels stored in the increasing order, starting with channel 1. Each stored speech frame starts with a ToC byte (see clause A.2.2.1.2). Note that no CMR byte is needed.

Non-received speech frames or frame-blocks between SID frames during non-speech periods shall be stored as NO_DATA frames. Frames or frame-blocks lost during transmission shall be stored as SPEECH_LOST frames in complete frame-blocks to keep synchronization with the original media.

A.3 Payload Format Parameters

A.3.1 EVS Media Type Registration

The media type for the EVS codec is to be allocated from the standards tree. This clause defines parameters of the EVS payload format. This media type registration covers real-time transfer via RTP and non-real-time transfers via stored files. All media type parameters defined in this Annex shall be supported. The receiver must ignore any unspecified parameter.

Media type name: audio

Media subtype name: EVS

Required parameters: none

Optional parameters:

The parameters defined below apply to RTP transfer only.

The following parameters are applicable to both EVS Primary mode and EVS AMR-WB IO mode:

ptime: see RFC 4566 [27].

maxptime: see RFC 4566 [27].

evs-mode-switch: Permissible values are 0 and 1. If evs-mode-switch is 0 or not present, EVS primary mode is used at the start or update of the session for the send and the receive directions. If evs-mode-switch is 1, EVS AMR-WB IO mode is used at the start or update of the session for the send and the receive directions.

hf-only: Permissible values are 0 and 1. If hf-only is 0 or not present, both Compact and Header-Full formats can be used in the session for the send and the receive directions. If hf-only is 1, only Header-Full format without zero padding for size collision avoidance is used.

dtx: Permissible values are 0 and 1. If dtx is 0, DTX is disabled in the session for the send and the receive directions. If dtx is 1 or not present, DTX is enabled. If dtx is included, dtx-recv is redundant but if dtx-recv is included, it shall be identical to dtx.

NOTE 1: If dtx is not present, DTX can still be disabled by the inclusion of dtx-recv=0 for the direction indicated by dtx-recv. See also clause A.3.3.1 and clause A.3.3.3.

dtx-recv: Permissible values are 0 and 1. If dtx-recv=0 is included for a payload type in the received SDP offer or the received SDP answer, and the payload type is accepted, the receiver shall disable DTX for the send direction. If dtx-recv=1 is included for a payload type in the received SDP offer or the received SDP answer, and this payload type is accepted, or if dtx-recv is not present for an accepted payload type, DTX is enabled.

NOTE 2: dtx-recv only applies for the media direction towards the SDP sender. If dtx-recv is not present, dtx determines if DTX is enabled or disabled. See also clause A.3.3.1 and clause A.3.3.3.

max-red: See RFC 4867 [15].

channels: The number of audio channels. See RFC 3551 [38]. If channels is not present, its default value is 1. If both ch-send and ch-recv are included in the SDP with different numbers of channels for sending and receiving directions, channels is set to the larger of the two parameters.

cmr: Permissible values are -1, 0, and 1. If cmr is -1 and the session is in the EVS primary mode, CMR on the RTP payload header is disabled in the session. If cmr is -1 and the session is in the EVS AMR-WB IO mode, CMR in the CMR byte is restricted to the values of EVS AMR-WB IO bit-rates and NO_REQ as specified in Table A.3. If cmr is 0 or not present, the values of CMR specified in Table A.3 are enabled. If cmr is 1, CMR shall be present in each packet. CMR shall be compliant with the negotiated bit-rate and bandwidth media type attributes for EVS primary and EVS AMR-WB IO modes.

The following parameters are applicable only to EVS Primary mode:

br: Specifies the range of source codec bit-rate for EVS Primary mode (see Table 1 [2]) in the session, in kilobits per second, for the send and the receive directions. The parameter can either have: a single bit-rate (br1); or a hyphen-separated pair of two bit-rates (br1-br2). If a single value is included, this bit-rate, br1, is used. If a hyphen-separated pair of two bit-rates is included, br1 and br2 are used as the minimum bit-rate and the maximum bit-rate respectively. br1 shall be smaller than br2. br1 and br2 have a value from the set: 5.9, 7.2, 8, 9.6, 13.2, 16.4, 24.4, 32, 48, 64, 96, and 128. 5.9 represents the average bit-rate of source controlled variable bit rate (SC-VBR) coding, and 7.2, …, 128 represent the bit-rates of constant bit-rate source coding. Only bit-rates supporting at least one of the allowed audio bandwidth(s) shall be used in the session (see clause A.3.3.1). If br is not present, all bit-rates consistent with the negotiated bandwidth(s) are allowed in the session unless br-send or br-recv is present. If br is included, br-send or br-recv is redundant but if either br-send or br-recv, or both are included, they shall be identical to br. If br-send and br-recv are not identical, br shall not be used.

br-send: Specifies the range of source codec bit-rate for EVS Primary mode (see Table 1 [2]) in the session, in kilobits per second, for the send direction. The parameter can either have: a single bit-rate (br1); or a hyphen-separated pair of two bit-rates (br1-br2). If a single value is included, this bit-rate, br1, is used. If a hyphen-separated pair of two bit-rates is included, br1 and br2 are used as the minimum bit-rate and the maximum bit-rate respectively. br1 shall be smaller than br2. br1 and br2 have a value from the set: 5.9, 7.2, 8, 9.6, 13.2, 16.4, 24.4, 32, 48, 64, 96, and 128. 5.9 represents the average bit-rate of source controlled variable bit-rate (SC-VBR) coding, and 7.2, …, 128 represent the bit-rates of constant bit-rate source coding. Only bit-rates supporting at least one of the allowed audio bandwidth(s) shall be used in the session (see clause A.3.3.1). If br-send is not present, all bit-rates consistent with the negotiated bandwidth(s) are allowed in the session unless br is present.

br-recv: Specifies the range of source codec bit-rate for EVS Primary mode (see Table 1 [2]) in the session, in kilobits per second, for the receive direction. The parameter can either have: a single bit-rate (br1); or a hyphen-separated pair of two bit-rates (br1-br2). If a single value is included, this bit-rate, br1, is used. If a hyphen-separated pair of two bit-rates is included, br1 and br2 are used as the minimum bit-rate and the maximum bit-rate respectively. br1 shall be smaller than br2. br1 and br2 have a value from the set: 5.9, 7.2, 8, 9.6, 13.2, 16.4, 24.4, 32, 48, 64, 96, and 128. 5.9 represents the average bit-rate of source controlled variable bit-rate (SC-VBR) coding, and 7.2, …, 128 represent the bit-rates of constant bit-rate source coding. Only bit-rates supporting at least one of the allowed audio bandwidth(s) shall be used in the session (see clause A.3.3.1). If br-recv is not present, all bit-rates consistent with the negotiated bandwidth(s) are allowed in the session unless br is present.

bw: Specifies the audio bandwidth for EVS Primary mode (see Table 1 [2]) to be used in the session for the send and the receive directions. bw has a value from the set: nb, wb, swb, fb, nb-wb, nb-swb, and nb-fb. nb, wb, swb, and fb represent narrowband, wideband, super-wideband, and fullband respectively, and nb-wb, nb-swb, and nb-fb represent all bandwidths from narrowband to wideband, super-wideband, and fullband respectively. If bw is not present, all bandwidths consistent with the negotiated bit-rate(s) are allowed in the session unless bw-send or bw-recv is present. If bw is included, bw-send or bw-recv is redundant but if either bw-send or bw-recv, or both are included, they shall be identical to bw. If bw-send and bw-recv are not identical, bw shall not be used.

bw-send: Specifies the bandwidth (see Table 1 [2]) to be used in the session for the send direction. bw-send has a value from the set: nb, wb, swb, fb, nb-wb, nb-swb, and nb-fb. nb, wb, swb, and fb represent narrowband, wideband, super-wideband, and fullband respectively, and nb-wb, nb-swb, and nb-fb represent all bandwidths from narrowband to wideband, super-wideband, and fullband respectively. If bw-send is not present, all bandwidths consistent with the negotiated bit-rate(s) are allowed in the session unless bw is present.

bw-recv: Specifies the bandwidth (see Table 1 [2]) to be used in the session for the receive direction. bw-recv has a value from the set: nb, wb, swb, fb, nb-wb, nb-swb, and nb-fb. nb, wb, swb, and fb represent narrowband, wideband, super-wideband, and fullband respectively, and nb-wb, nb-swb, and nb-fb represent all bandwidths from narrowband to wideband, super-wideband, and fullband respectively. If bw-recv is not present, all bandwidths consistent with the negotiated bit-rate(s) are allowed in the session unless bw is present.

ch-send: Specifies the number of audio channels to be used in the session for the send direction. ch-send has an integer value from 1 to the maximum number of audio channels (see also clause A.3.2). If ch-send is not present, ch-send=1, mono, is supported.

ch-recv: Specifies the number of audio channels to be used in the session for the receive direction. ch-recv has an integer value from 1 to the maximum number of audio channels (see also clause A.3.2). If ch-recv is not present, ch-recv=1, mono, is supported.

ch-aw-recv: Specifies how channel-aware mode is configured or used for the receive direction. Permissible values are -1, 0, 2, 3, 5, and 7. If ch-aw-recv is -1, channel-aware mode is disabled in the session for the receive direction. If ch-aw-recv is 0 or not present, partial redundancy (channel-aware mode) is not used at the start of the session for the receive direction. If ch-aw-recv is positive (2, 3, 5, or 7), partial redundancy (channel-aware mode) is used at the start of the session for the receive direction using the value as the offset (See NOTE below). Partial redundancy is supported only when the bit-rate is 13.2 kbps and the bandwidth is wb or swb.

NOTE 3: If a positive (2, 3, 5, or 7) value of ch-aw-recv is declared for a payload type and the payload type is accepted, the receiver of the parameter shall send partial redundancy (channel-aware mode) with the value of ch-aw-recv as the offset when operating at 13.2 kbit/s in the session . Note that if ch-aw-recv=-1 is not declared for a payload type and the payload type is accepted, the value of ch-aw-recv may be modified during the session by an adaptation request.

If ch-aw-recv=0 is declared or not present for a payload type and the payload type is accepted, the receiver of the parameter shall not send partial redundancy (channel-aware mode) at the start of the session.

If ch-aw-recv=-1 is declared for a payload type and the payload type is accepted, the receiver of the parameter shall not send partial redundancy (channel-aware mode) in the session.

If ch-aw-recv is not present or a non-negative (0, 2, 3, 5, or 7) value of ch-aw-recv is declared for a payload type and the payload type is accepted, partial redundancy (channel-aware mode) can be activated or deactivated during the session based on the expected or estimated channel condition through adaptation signaling, such as CMR (see Annex A.2) or RTCP based signaling (see clause 10.2 of [13]).

If ch-aw-recv is not present or a non-negative (0, 2, 3, 5, or 7) value of ch-aw-recv is declared for a payload type and the payload type is accepted, the partial redundancy offset value can also be adjusted during the session based on the expected or estimated channel condition through adaptation signaling.

NOTE 4: The frame erasure rate indicator for the channel-aware mode has two permissible values (LO, HI) and this indicator has to be initialized to HI, as specified in clause 5.8.4.

The following parameters are applicable only to EVS AMR-WB IO mode:

mode-set: Restricts the active codec mode set to a subset of all modes when the EVS codec operates in AMR-WB IO, for example, to be able to support transport channels such as GSM or UMTS networks. Possible value is a comma-separated list of modes from the set: 0, …, 8 (see Table 1a [36]). If mode-set is specified, it must be abided, and frames encoded with AMR-WB IO outside of the subset must not be sent in any RTP payload or used in codec mode request signal. If not present, all codec modes of AMR-WB IO are allowed for the payload type.

mode-change-period: See RFC 4867 [15].

mode-change-capability: See RFC 4867 [15], except that the default and the only allowed value of mode-change-capability is 2 for EVS AMR-WB IO. As the default and the only allowed value of mode-change-capibility is 2 in EVS AMR-WB IO, it is not required to include this parameter in the SDP.

mode-change-neighbor: See RFC 4867 [15].

Optional parameters of AMR-WB (see clause 8.2 of [15]) not defined above shall not be used in the EVS AMR-WB IO mode.

A.3.2 Mapping Media Type Parameters into SDP

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [27], which is commonly used to describe RTP sessions. When SDP is used to specify sessions employing the EVS codec, the mapping is as follows:

- The media type ("audio") goes in SDP "m=" as the media name.
- The media subtype (payload format name) goes in SDP "a=rtpmap" as the encoding name. The RTP clock rate in "a=rtpmap" shall be 16000, and the encoding parameters (number of channels) shall either be explicitly set to N or omitted, implying a default value of 1. The values of N that are allowed are specified in Section 4.1 in [38]. If ch-send and/or ch-recv paramaters are supplied, the number of channels N shall be the larger value given in those parameters.
- The parameters "ptime" and "maxptime" go in the SDP "a=ptime" and "a=maxptime" attributes, respectively.
- Any remaining parameters go in the SDP "a=fmtp" attribute by copying them directly from the media type parameter string as a semicolon-separated list of parameter=value pairs.

Mapping to fields in SDP is specified in clause 6 of [13].

A.3.3 Detailed Description of Usage of SDP Parameters

A.3.3.1 Offer-Answer Model Considerations

The following considerations apply when using SDP Offer-Answer procedures to negotiate the use of EVS payload in RTP:

dtx: When dtx is not offered, i.e., not included, for a payload type, the answerer may include dtx for the payload type in the SDP answer. When dtx is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove dtx for the payload type in the SDP answer. When dtx-recv is offered and the answerer includes dtx, the value of dtx in the answer shall be identical to the value of dtx-recv in the offer.
When dtx is not present in the SDP answer (and thus was not present in the SDP offer), the following applies:

- If dtx-recv is not present in the SDP offer, DTX shall be enabled at least in the direction towards the offerer.
- If dtx-recv is present in the SDP offer, DTX shall be enabled or disabled towards the offerer depending on the value of dtx-recv in the offer.
- If dtx-recv is not present in the SDP answer, DTX shall be enabled at least in the direction towards the answerer.
- If dtx-recv is present in the SDP answer, DTX shall be enabled or disabled towards the answerer depending on the value of dtx-recv in the answer.

dtx-recv: The answerer may include dtx-recv for the payload type in the SDP answer irrespective of the presence and value of dtx-recv in the offer.

hf-only: When hf-only is not offered for a payload type, the answerer may include hf-only for the payload type in the SDP answer. When hf-only is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove hf-only for the payload type in the SDP answer.

evs-mode-switch: When evs-mode-switch is not offered for a payload type, the answerer may include evs-mode-switch for the payload type in the SDP answer. When evs-mode-switch is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove evs-mode-switch for the payload type in the SDP answer.

br: When the same bit-rate or bit-rate range is defined for the send and the receive directions, br should be used but br-send and br-recv may also be used. br can be used even if the session is negotiated to be sendonly, recvonly, or inactive. For sendonly session, br and br-send can be interchangeably used. For recvonly session, br and br-recv can be interchangeably used. When br is not offered for a payload type, the answerer may include br for the payload type in the SDP answer. When br is offered for a payload type and the payload type is accepted, the answerer shall include br in the SDP answer which shall be identical to or a subset of br for the payload type in the SDP offer.

br-send: When br-send is not offered for a payload type, the answerer may include br-recv for the payload type in the SDP answer. When br-send is offered for a payload type and the payload type is accepted, the answerer shall include br-recv in the SDP answer, and the br-recv shall be identical to or a subset of br-send for the payload type in the SDP offer.

br-recv: When br-recv is not offered for a payload type, the answerer may include br-send for the payload type in the SDP answer. When br-recv is offered for a payload type and the payload type is accepted, the answerer shall include br-send in the SDP answer, and the br-send shall be identical to or a subset of br-recv for the payload type in the SDP offer.

bw: When the same bandwidth or bandwidth range is defined for the send and the receive directions, bw should be used but bw-send and bw-recv may also be used. bw can be used even if the session is negotiated to be sendonly, recvonly, or inactive. For sendonly session, bw and bw-send can be interchangeably used. For recvonly session, bw and bw-recv can be interchangeably used. When bw is not offered for a payload type, the answerer may include bw for the payload type in the SDP answer. When bw is offered for a payload type and the payload type is accepted, the answerer shall include bw in the SDP answer, which shall be identical to or a subset of bw for the payload type in the SDP offer.

bw-send: When bw-send is not offered for a payload type, the answerer may include bw-recv for the payload type in the SDP answer. When bw-send is offered for a payload type and the payload is accepted, the answerer shall include bw-recv in the SDP answer, and the bw-recv shall be identical to or a subset of bw-send for the payload type in the SDP offer.

bw-recv When bw-recv is not offered for a payload type, the answerer may include bw-send for the payload type in the SDP answer. When bw-recv is offered for a payload type and the payload is accepted, the answerer shall include bw-send in the SDP answer, and the bw-send shall be identical to or a subset of bw-recv for the payload type in the SDP offer.

cmr: When cmr is not offered for a payload type, the answerer may include cmr for the payload type in the SDP answer. When cmr is offered for a payload type and the payload type is accepted, the answerer shall not modify or remove cmr for the payload type in the SDP answer.

channels See <encoding parameters> of a=rtpmap attribute specified in RFC 4566 [27]. If ch-send and ch-recv are offered for a payload type with different numbers of channels for sending and receiving directions, channels is set to the larger of the two parameters.

ch-send: When ch-send is offered for a payload type and the payload type is accepted, the answerer shall include ch-recv in the SDP answer, and the ch-recv shall be identical to the ch-send parameter for the payload type in the SDP offer.

ch-recv When ch-recv is offered for a payload type and the payload type is accepted, the answerer shall include ch-send in the SDP answer, and the ch-send shall be identical to the ch-recv parameter for the payload type in the SDP offer.

When a single bit-rate is offered, the answerer may accept the offered bit-rate or reject the offered bit-rate. If the offered bit-rate is accepted, this bit-rate shall be used also in the SDP answer. If the offered bit-rate is accepted but the session is changed from sendrecv to sendrecv or recvonly, the offered bit-rate shall be used in the br, br-send or br-recv parameter included in the SDP answer. Otherwise, the RTP payload type shall be rejected.

When a bit-rate range is offered, the answerer: may accept the offered bit-rate range, modify the offered bit-rate range, select a single bit-rate, or may reject the offered bit-rate range. Otherwise, the RTP payload type shall be rejected.

When an offered bit-rate range is modified for the answer, the following rules apply:

- The lower bit-rate limit 'br1' can be kept unchanged or can be increased up to 'br2', but cannot be decreased.
- The upper bit-rate limit 'br2' can be kept unchanged or can be decreased down to 'br1', but cannot be increased.

When an offered bit-rate range is answered with a single bit-rate, this bit-rate shall be one of the offered bit-rates.

Rejecting all RTP payload types may lead to rejecting the media type and possibly even the whole SIP INVITE.

The bit-rates and bandwidths indicated in the negotiated media type attributes shall be consistent with Table A.6. Each 'x' represents a bit-rate and bandwidth combination supported by the EVS codec.

Table A.6: Allowed bit-rates and audio bandwidths

	5.9	7.2	8	9.6	13.2	16.4	24.4	32	48	64	96	128
nb	x	x	x	x	x	x	x					
wb	x	x	x	x	x	x	x	x	x	x	x	x
swb				x	x	x	x	x	x	x	x	x
fb						x	x	x	x	x	x	x

If no bit rate parameter and no bandwidth parameter are specified, all bit-rates and bandwidths combinations as specified in Table A.6 are allowed in the session.

A.3.3.2 Examples

SDP offer/answer procedure examples for MTSI are in A.14 of [13].

Setting up a symmetric dual-mono session in both sending and receiving direction, can be done with SDP offer and SDP answer negotiating the same number of channels on the 'a=rtpmap' line in the SDP offer and SDP answer. An example SDP offer/answer negotiation for using the same number of channels for sending and receiving directions is included below:

Example SDP offer

```
m=audio 49152 RTP/AVP 96 97 98 99 100 101 102 103
a=rtpmap:96 EVS/16000/2
a=fmtp:96 br=16.4; bw=nb-swb; max-red=220
a=rtpmap:97 EVS/16000/1
a=fmtp:97 br=13.2-24.4; bw=nb-swb; max-red=220
a=rtpmap:98 AMR-WB/16000/2
a=fmtp:98 mode-change-capability=2; max-red=220
a=rtpmap:99 AMR-WB/16000/2
a=fmtp:99 mode-change-capability=2; max-red=220; octet-align=1
a=rtpmap:100 AMR-WB/16000/1
a=fmtp:100 mode-change-capability=2; max-red=220
a=rtpmap:101 AMR-WB/16000/1
a=fmtp:101 mode-change-capability=2; max-red=220; octet-align=1
a=rtpmap:102 AMR/8000/1
a=fmtp:102 mode-change-capability=2; max-red=220
a=rtpmap:103 AMR/8000/1
a=fmtp:103 mode-change-capability=2; max-red=220; octet-align=1
a=ptime:20
a=maxptime:240
```

Example SDP answer

```
m=audio 49152 RTP/AVP 96
a=rtpmap:96 EVS/16000/2
a=fmtp:96 br=16.4; bw=nb-swb; max-red=220
a=ptime:20
a=maxptime:240
```

It is possible to use one m= line when setting up a session with equal number of channels in both directions.

Setting up a session with asymmetric number of channels for different directions is possible by negotiating different number of channels using the ‘ch-send=<#>’ and the ‘ch-recv=#’ parameters.

A.3.3.3 Interactions of the dtx and dtx-recv parameters

Table A.7 lists all allowed combinations of the dtx and dtx-recv parameters in SDP offers and answers, and their meaning. Combinations of the dtx and dtx-recv parameters in SDP offers and answers not contained in Table A.7 shall not be used; the error handling if such combinations are encountered is left to the implementation.

Table A.7: Allowed combinations of the dtx and dtx-recv parameter in SDP offer and answer

Number	SDP offer		SDP answer		DTX towards offerer enabled?	DTX towards answerer enabled?
	dtx	dtx recv	dtx	dtx recv		
1	-	-	-	-	y	y
2	-	0	-	-	n	y
3	-	1	-	-	y	y
4	-	-	0	-	n	n
5	0	-	0	-	n	n
6	-	0	0	-	n	n
7	0	0	0	-	n	n
8	-	-	1	-	y	y
9	1	-	1	-	y	y
10	-	1	1	-	y	y
11	1	1	1	-	y	y
12	-	-	-	0	y	n
13	-	0	-	0	n	n
14	-	1	-	0	y	n
15	-	-	0	0	n	n
16	0	-	0	0	n	n
17	-	0	0	0	n	n
18	0	0	0	0	n	n
19	-	-	-	1	y	y
20	-	0	-	1	n	y
21	-	1	-	1	y	y
22	-	-	1	1	y	y
23	1	-	1	1	y	y
24	-	1	1	1	y	y
25	1	1	1	1	y	y

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2014-09	SP-65	SP-140460				Presented to TSG SA#65 for approval	1.0.0
2014-09	SP-65					Approved at TSG SA#65	12.0.0
2014-12	SP-66	SP-140726	0001	3		Corrections to Algorithmic Description Text	12.1.0
2014-12	SP-66	SP-140726	0002	3		Incorporating RTP Payload Format and Media Type Parameters	12.1.0
2015-03	SP-67	SP-150086	0003	1		Corrections to the Algorithmic and the RTP Payload Format Descriptions	12.2.0
2015-04						Editorial Corrections (date and version number in the headings of each multi-part files)	12.2.1
2015-06	SP-68	SP-150203	0004	1		Corrections to the Algorithmic Description	12.3.0
2015-09	SP-69	SP-150434	0005	1		Corrections to the Algorithmic Description	12.4.0
2015-09	SP-69	SP-150434	0006	4		Corrections to Payload Format Parameters	12.4.0
2015-12	SP-70	SP-150639	0007	1		Corrections to the Algorithmic Description	12.5.0
2015-12	SP-70	SP-150639	0008	-		Handling Received CMR	12.5.0
2015-12	SP-70			-		Version for Release 13	13.0.0
2016-03	SP-71	SP-160064	0013	1		Correction of mode-change-capability and channel-aware configuration	13.1.0
2016-06	72	SP-160257	0015		A	Corrections to the Algorithmic Description	13.2.0
2016-06	72	SP-160257	0017	1	A	Corrections to CMR Handling for AMR-WB IO mode	13.2.0
2016-06	72	SP-160257	0019	2	A	EVS-CMR-Only packets	13.2.0
2016-09	73	SP-160589	0022	1	A	Corrections to the Algorithmic Description	13.3.0
2016-09	73	SP-160589	0023	2	A	Give "NO_REQ" and "none" a clear definition	13.3.0
2016-09	73	SP-160589	0024	-	A	Corrections regarding the EVS dtx and dtx-recv MIME parameters	13.3.0
2016-12	74	SP-160770	0027	1	A	Corrections to the Algorithmic Description	13.4.0
2016-12	74	SP-160770	0029	-	A	Clarifications for EVS Rate and Mode Control	13.4.0

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-03	75					Version for Release 14	14.0.0
2017-06	76	SP-170316	0032	-	A	Corrections to the Algorithmic Description	14.1.0
2017-12	78	SP-170820	0035	1	A	Corrections to the Algorithmic Description	14.2.0
2017-12	78	SP-170822	0036	-	F	Handling of hf-only parameter	14.2.0
2018-06	80					Version for Release 15	15.0.0
2018-12	82	SP-180965	0037	1	A	Corrections to the Algorithmic Description	15.1.0
2019-03	83	SP-190031	0045	1	A	Correction of EVS SID update	15.2.0
2019-06	84	SP-190338	0046	-	B	Correction and addition of reference to Alt_FX_EVS implementation	16.0.0
2020-06	SA#88-e	SP-200386	0051		A	Corrections of algorithmic description	16.1.0
2020-06	SA#88-e	SP-200396	0052	2	F	Corrections of ch-aw-recv specification	16.1.0
2020-10	Post SA#88-e					Editorial: Corrections in Change History table	16.1.1
2021-12	SA#94-e	SP-211345	0057	1	F	Correction and addition to TS 26.445 specification	16.2.0
2022-04	SA#95-e					Upgraded to Release 17	17.0.0