Presentation of Specification to TSG

Presentation to:	**TSG SA Meeting #10**
Document for presentation:	**TS 26.194, Version 1.0.0**
Presented for:	**Information**
Agenda Item:	**7.4.1**

Title of document:

AMR Wideband Speech Codec; Voice Activity Detector (VAD)

3G TS 26.194 V1.0.0 (2000-12)

Technical Specification

3rd Generation Partners
Technical Specification Group Services
Aspects;
Mandatory Speech Codec speech processing functions
AMR Wideband speech codec; Voice Activity Detector
(VAD)
(Release 4)

Adaptive Multi-Rate, Wideband speech codec

Keywords

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

3GPP

Contents

Foreword

This Technical Specification has been produced by the 3GPP.

The present document defines an error concealment procedure, also termed frame substitution and muting procedure, of the wideband telephony speech service employing the Adaptive Multi-Rate – Wideband (AMR-WB) speech coder within the 3GPP system.

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

Version x.y.z

where:

x   the first digit:

1   presented to TSG for information;

2   presented to TSG for approval;

3   Indicates TSG approved document under change control.

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

z   the third digit is incremented when editorial only changes have been incorporated in the specification;

1 Scope

This document specifies the Voice Activity Detector (VAD) to be used in the Discontinuous Transmission (DTX) as described in [3].

The requirements are mandatory on any VAD to be used either in User Equipment (UE) or Base Station Systems (BSS)s that utilize the AMR wideband speech codec.

2 Normative References

This TS incorporates by dated and undated reference, provisions from other publications. These normative references are cited in the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this TS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1]   TS 26.173: "ANSI-C code for the Adaptive Multi Rate wideband speech codec" .

[2]   TS 26.190: "AMR wideband Speech Codec Speech Transcoding Functions" .

[3]   TS 26.193: "AMR wideband Speech codec; Source Controlled Rate Operation".

[4]   ITU, The International Telecommunications Union, Blue Book, Vol. III, Telephone Transmission Quality, IXth Plenary Assembly, Melbourne, 14-25 November, 1988, Recommendation G.711, Pulse code modulation (PCM) of voice frequencies.

3 Technical Description

3.1 Definitions, symbols and abbreviations

3.1.1 Definitions

For the purposes of this TS, the following definitions apply:

frame: Time interval of 20 ms corresponding to the time segmentation of the speech transcoder.

3.1.2 Symbols

For the purposes of this TS, the following symbols apply.

3.1.2.1 Variables

bckr_est[n]   background noise estimate at the frequency band "n"

burst_count   counts length of a speech burst, used by VAD hangover addition

hang_count   hangover counter, used by VAD hangover addition

level[n]   signal level at the frequency band "n"

new_speech   pointer of the speech encoder, points a buffer containing last received samples of a speech frame [2]

noise_level   estimated noise level

pow_sum   input power

s(i)   samples of the input frame

snr_sum   measure between input frame and noise estimate

speech_level estimated speech level

stat_count   stationary counter

stat_rat   measure indicating stationary of the input frame

tone_flag   flag indicating the presence of a tone

vad_thr   VAD threshold

VAD_flag   Boolean VAD flag

vadreg   intermediate VAD decision

3.1.2.2   Constants

ALPHA_UP1	constant for updating noise estimate (see subclause 3.3.5.2)
ALPHA_DOWN1	constant for updating noise estimate (see subclause 3.3.5.2)
ALPHA_UP2	constant for updating noise estimate (see subclause 3.3.5.2)
ALPHA_DOWN2	constant for updating noise estimate (see subclause 3.3.5.2)
ALPHA3	constant for updating noise estimate (see subclause 3.3.5.2)
ALPHA4	constant for updating average signal level (see subclause 3.3.5.2)
ALPHA5	constant for updating average signal level (see subclause 3.3.5.2)
BURST_HIGH	constant for controlling VAD hangover addition (see subclause 3.3.5.1)
BURST_P1	constant for controlling VAD hangover addition (see subclause 3.3.5.1)
BURST_SLOPE	constant for controlling VAD hangover addition (see subclause 3.3.5.1)
COEFF3	coefficient for the filter bank (see subclause 3.3.1)
COEFF5_1	coefficient for the filter bank (see subclause 3.3.1)
COEFF5_2	coefficient for the filter bank (see subclause 3.3.1)
HANG_HIGH	constant for controlling VAD hangover addition (see subclause 3.3.5.1)
HANG_LOW	constant for controlling VAD hangover addition (see subclause 3.3.5.1)
HANG_P1	constant for controlling VAD hangover addition (see subclause 3.3.5.1)

HANG_SLOPE constant for controlling VAD hangover addition (see subclause 3.3.5.1)

FRAME_LEN size of a speech frame, 256 samples (20 ms)

MIN_SPEECH_LEVEL1 constant for speech estimation (see subclause 3.3.5.3)

MIN_SPEECH_LEVEL2 constant for speech estimation (see subclause 3.3.5.3)

MIN_SPEECH_SNR constant for VAD threshold adaptation (see subclause 3.3.5)

NO_P1 constant for VAD threshold adaptation (see subclause 3.3.5)

NO_SLOPE constant for VAD threshold adaptation (see subclause 3.3.5)

NOISE_MAX maximum value for noise estimate (see subclause 3.3.5.2)

NOISE_MIN minimum value for noise estimate (see subclause 3.3.5.2)

POW_TONE_THR threshold for tone detection (see subclause 3.3.5)

SP_ACTIVITY_COUNT constant for speech estimation (see subclause 3.3.5.3)

SP_ALPHA_DOWN constant for speech estimation (see subclause 3.3.5.3)

SP_ALPHA_UP constant for speech estimation (see subclause 3.3.5.3)

SP_CH_MAX constant for VAD threshold adaptation (see subclause 3.3.5)

SP_CH_MIN constant for VAD threshold adaptation (see subclause 3.3.5)

SP_EST_COUNT constant for speech estimation (see subclause 3.3.5.3)

SP_P1 constant for VAD threshold adaptation (see subclause 3.3.5)

SP_SLOPE constant for VAD threshold adaptation (see subclause 3.3.5)

STAT_COUNT threshold for stationary detection (see subclause 3.3.5.2)

STAT_THR threshold for stationary detection (see subclause 3.3.5.2)

STAT_THR_LEVEL threshold for stationary detection (see subclause 3.3.5.2)

THR_HIGH constant for VAD threshold adaptation (see subclause 3.3.5)

TONE_THR threshold for tone detection (see subclause 3.3.3)

VAD_POW_LOW constant for controlling VAD hangover addition (see subclause 3.3.5.1)

3.1.2.3 Functions

+	Addition
-	Subtraction
*	Multiplication
/	Division
| x |	absolute value of x
AND	Boolean AND
OR	Boolean OR

$$\sum_{n=a}^{b} x(n) \qquad = x(a) + x(a+1) + \ldots + x(b-1) + x(b)$$

$$\textbf{MIN(x,y)} \qquad = \begin{cases} x, x \le y \\ y, y < x \end{cases}$$

$$\textbf{MAX(x,y)} \qquad = \begin{cases} x, x \ge y \\ y, y > x \end{cases}$$

3.1.3 Abbreviations

ANSI	American National Standards Institute
DTX	Discontinuous Transmission
VAD	Voice Activity Detector
CNG	Comfort Noise Generation

3.2 General

The function of the VAD algorithm is to indicate whether each 20 ms frame contains signals that should be transmitted, e.g. speech, music or information tones. The output of the VAD algorithm is a Boolean flag (VAD_flag) indicating presence of such signals.

3.3 Functional description

The block diagram of the VAD algorithm is depicted in figure 1. The VAD algorithm uses parameters of the speech encoder to compute the Boolean VAD flag (VAD_flag). This input frame for VAD is sampled at the 6.4 kHz frequency and thus it contains 256 samples. Samples of the input frame (s(i)) are divided into sub-bands and level of the signal (level[n]) in each band is calculated. Input for the tone detection function are the normalized open-loop pitch gains which are calculated by open-loop pitch analysis of the speech encoder. The tone detection function computes a flag (tone_flag) which indicates presence of a signalling tone, voiced speech, or other strongly periodic signal. Background noise level (bckr_est[n]) is estimated in each band based on the VAD decision, signal stationarity and the tone-flag. Intermediate VAD decision is calculated by comparing input SNR (level[n]/bckr_est[n]) to an adaptive threshold. The threshold is adapted based on noise and long term speech estimates. Finally, the VAD flag is calculated by adding hangover to the intermediate VAD decision.
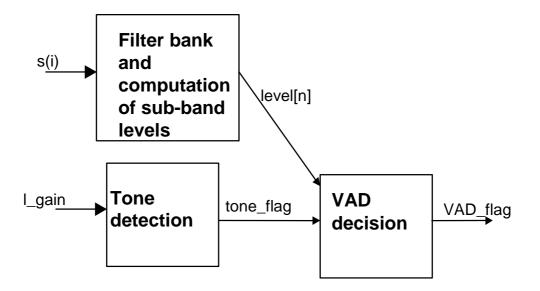
Figure 1. Simplified block diagram of the VAD algorithm

3.3.1   Filter bank and computation of sub-band levels

The input signal is divided into frequency bands using a 12-band filter bank (figure 2). Cut-off frequencies for the filter bank are shown in table 1.

Table 1. Cut-off frequencies for the filter bank

Band number	Frequencies
1	0 – 200 Hz
2	200 – 400 Hz
3	400 – 600 Hz
4	600 – 800 Hz
5	800 – 1200 Hz
6	1200 – 1600 Hz
7	1600 – 2000 Hz
8	2000 – 2400 Hz
9	2400 - 3200 Hz
10	3200 – 4000 Hz
11	4000 – 4800 Hz
12	4800 – 6400 Hz

Input for the filter bank is a speech frame pointed by the new_speech pointer of the speech encoder [1]. Input values for the filter bank are scaled down by one bit. This ensures safe scaling, i.e. saturation can not occur during calculation of the filter bank.
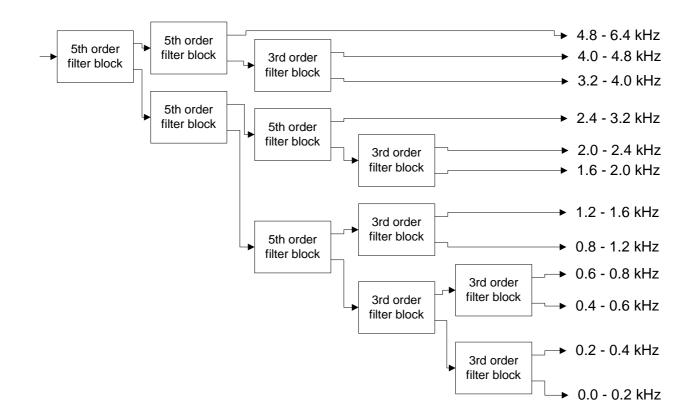
Figure 2. Filter bank

The filter bank consists of 5th and 3rd order filter blocks. Each filter block divides the input into high-pass and low-pass parts and decimates the sampling frequency by 2. The 5th order filter block is calculated as follows:

$$x_{lp}(i) = 0.5*(A_1(x(2*i)) + A_2(x(2*i+1))) \tag{1a}$$

$$x_{hp}(i) = 0.5*(A_1(x(2*i)) - A_2(x(2*i+1))) \tag{1b}$$

where

$x(i)$ input signal for a filter block

$x_{lp}(i)$ low-pass component

$x_{hp}(i)$ high-pass component

The 3rd order filter block is calculated as follows:

$$x_{lp}(i) = 0.5*(x(2*i+1) + A_3(x(2*i))) \tag{2a}$$

$$x_{hp}(i) = 0.5*(x(2*i+1) - A_3(x(2*i))) \tag{2b}$$

The filters $A_1()$, $A_2()$, and $A_3()$ are first order direct form all-pass filters, whose transfer function is given by:

$$A(z) = \frac{C + z^{-1}}{1 + C*z^{-1}}, \tag{3}$$

where C is the filter coefficient.

Coefficients for the all-pass filters $A_1()$, $A_2()$, and $A_3()$ are COEFF5_1, COEFF5_2, and COEFF3, respectively.

Signal level is calculated at the output of the filter bank at each frequency band as follows:

$$level(n) = \sum_{i=START_n}^{END_n} |x_n(i)|, \qquad (4)$$

where:

n   index for the frequency band

$x_n(i)$ sample i at the output of the filter bank at frequency band n

$$START_n = \begin{cases} -6, & 1 \le n \le 4 \\ -12, & 5 \le n \le 8 \\ -24, & 9 \le n \le 11 \\ -48, & n = 12 \end{cases}$$

$$END_n = \begin{cases} 7, & 1 \le n \le 4 \\ 15, & 5 \le n \le 8 \\ 31, & 9 \le n \le 11 \\ 63, & n = 12 \end{cases}$$

Negative indices of $x_n(i)$ refer to the previous frame.

3.3.2   Tone detection

The purpose of the tone detection function is to detect information tones, vowel sounds and other periodic signals. The tone detection uses normalized open-loop pitch gains (ol_gain), which are received from the speech encoder. If the pitch gain is higher than the constant TONE_THR, tone is detected and the tone flag is set:

if (ol_gain > TONE_THR)

      tone_flag = 1

The open-loop pitch search and correspondingly the tone flag is computed twice in each frame, except for mode 6.60 kbit/s, where it is computed only once.

3.3.3   VAD decision

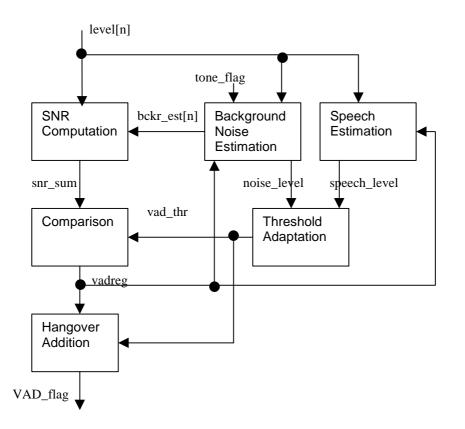The block diagram of the VAD decision algorithm is shown in figure 3.

Figure 3. Simplified block diagram of the VAD decision algorithm

Power of the input frame is calculated as follows:

$$frame\_pow = \sum_{i=0}^{FRAME\_LEN} s(i) * s(i), \tag{5}$$

where samples s(i) of the input frame are pointed by the new_speech pointer of the speech encoder. Variable pow_sum is sum of the powers of the current and previous frames. If pow_sum is lower than the constant POW_TONE_THR, tone-flag is set to zero.

The difference between the signal levels of the input frame and the background noise estimate is calculated as follows:

$$snr\_sum = \sum_{n=1}^{12} MAX(1.0, \frac{level[n]}{bckr\_est[n]})^2 , \tag{6}$$

where:

level[n]   signal level at band n

bckr_est[n]   level of background noise estimate at band n

VAD decision is made by comparing the variable snr_sum to a threshold. The threshold (vad_thr) is adapted to get desired sensitivity depending on estimated speech and background noise levels.

Average background noise level is calculated by adding noise estimates at each band except the lowest band:

$$noise\_level = \sum_{n=2}^{12} bckr\_est[n]$$

(7)

If SNR is lower that the threshold (MIN_SPEECH_SNR), speech level is increased as follows:

If (speech_level/noise_level < MIN_SPEECH_SNR)

Speech_level = MIN_SPEECH_SNR * noise_level

Logarithmic value for noise estimate is calculated as follows:

$$i\log 2\_noise\_level = \log_2 (noise\_level)$$ (8)

Before logarithmic value from the speech estimate is calculated, MIN_SPEECH_SNR*noise_level is subtracted from the speech level to correct its value in low SNR situations.

$$i\log 2\_speech\_level = \log_2 (speech\_level - MIN\_SPEECH\_SNR* noise\_level)$$ (9)

Threshold for VAD decision is calculated as follows:

Vad_thr = NO_SLOPE * (ilog2_noise_level - NO_P1) + THR_HIGH + MIN(SP_CH_MAX, MAX(SP_CH_MIN, SP_CH_MIN + SP_SLOPE * (ilog2_speech_level – SP_P1))),   (10)

where NO_SLOPE, SP_SLOPE, NO_P1, SP_P1, THR_HIGH, SP_CH_MAX and SP_CH_MIN are constants.

The variable vadreg indicates intermediate VAD decision and it is calculated as follows:

```
if (snr_sum > vad_thr)
    vadreg = 1
else
    vadreg = 0
```

3.3.3.1   Hangover addition

Before the final VAD flag is given, a hangover is added. The hangover addition helps to detect low power endings of speech bursts, which are subjectively important but difficult to detect.

VAD flag is set to "1" if less that hang_len frames with "0" decision have been elapsed since burst_len consecutive "1" decisions have been detected. The variables hang_len and burst_len are computed using vad_thr as follows:

hang_len = MAX(HANG_LOW, (HANG_SLOPE * (vad_thr – HANG_P1) + HANG_HIGH))   (11)

burst_len = BURST_SLOPE * (vad_thr – BURST_P1) + BURST_HIGH)   (12)

The power of the input frame is compared to a threshold (VAD_POW_LOW). If the power is lower, the VAD flag is set to "0" and no hangover is added. The VAD_flag is calculated as follows:

```
Vad_flag = 0;
if (pow_sum < VAD_POW_LOW)
    burst_count = 0
    hang_count = 0
else
    if (vadreg = 1)
        burst_count = burst_count + 1
        if (burst_count >= burst_len)
            hang_count = hang_len
        VAD_flag = 1
    else
        burst_count = 0
```

```
if (hang_count > 0)
    hang_count = hang_count - 1
    VAD_flag=1
```

3.3.3.2   Background noise estimation

Background noise estimate (bckr_est[n]) is updated using amplitude levels of the previous frame. Thus, the update is delayed by one frame to avoid undetected start of speech bursts to corrupt the noise estimate. The update speed is for the current frame is selected using intermediate VAD decisions (vadreg) and stationarity counter (stat_count) as follows:

```
if (vadreg for the last 4 frames has been zero)
    alpha_up = ALPHA_UP1
    alpha_down = ALPHA_DOWN1
else if (stat_count = 0)
    alpha_up = ALPHA_UP2
    alpha_down = ALPHA_DOWN2
else
    alpha_up = 0
    alpha_down = ALPHA3
```

The variable stat_count indicates stationary and its propose is explained later in this subclause. The variables alpha_up and alpha_down define the update speed for upwards and downwards, respectively. The update speed for each band "n" is selected as follows:

$$if\ (bckr\_est_m[n] < level_{m-1}[n])$$
```
    alpha[n] = alpha_up
else
    alpha[n] = alpha_down
```

Finally, noise estimate is updated as follows:

$$bckr\_est_{m+1}[n] = (1.0 - alpha[n]) * bckr\_est_m[n] + alpha[n] * level_{m-1}[n], \quad (13)$$

where:

      n   index of the frequency band

    m   index of the frame

Level of the background estimate (bckr_est[n]) is limited between constants NOISE_MIN and NOISE_MAX.

If level of background noise increases suddenly, vadreg will be set to "1" and background noise is not normally updated upwards. To recover from this situation, update of the background noise estimate is enabled if the intermediate VAD decision (vadreg) is "1" for enough long time and spectrum is stationary. Stationary (stat_rat) is estimated using following equation:

$$stat\_rat = \sum_{n=1}^{12} \frac{MAX(STAT\_THR\_LEVEL, MAX(ave\_level_m[n], level_m[n]))}{MAX(STAT\_THR\_LEVEL, MIN(ave\_level_m[n], level_m[n]))}, \quad (14)$$

where:

      STAT_THR_LEVEL   a constant

      n   index of the frequency band

    m   index of the frame

      ave_level   average level of the input signal

If the stationary estimate (stat_rat) is higher than a threshold, the stationary counter (stat_count) is set to the initial value defined by constant STAT_COUNT. If the signal is not stationary but speech has been detected (VAD decision is "1"), stat_count is decreased by one in each frame until it is zero.

```
if (5 last tone flags have been one)
    stat_count = STAT_COUNT
else
    if (8 last internal VAD decisions have been zero) OR (stat_rat > STAT_THR)
        stat_count = STAT_COUNT
    else
        if (vadreg) AND (stat_count ≠ 0)
            stat_count = stat_count – 1
```

The average signal levels (ave_level[n]) are calculated as follows:

$$ave\_level_{m+1}[n] = (1.0 - alpha) * ave\_level_m[n] + alpha * level_m[n] \tag{15}$$

The update speed (alpha) for the previous equation is selected as follows:

```
if (stat_count = STAT_COUNT)
    alpha = 1.0
else if (vadreg = 1)
    alpha=ALPHA5
else
    alpha = ALPHA4
```

3.3.3.3   Speech level estimation

First, full-band input level is calculated by summing input levels in each band except the lowest band as follows:

$$in\_level = \sum_{n=2}^{12} level[n] \tag{16}$$

A frame is assumed to contain speech if its level if high enough (MIN_SPEECH_LEVEL1), and the intermediate VAD flag (vadreg) is set or the input level is higher than the current speech level estimate. Maximum level (sp_max) from SP_EST_COUNT frames is searched. If the SP_ACTIVITY_COUNT number of speech frames is located in within SP_EST_COUNT number of frames, speech level estimate is updated by the maximum signal level (sp_max). The pseudocode for the speech level estimation is as follows:

```
If  (SP_ACTIVITY_COUNT > SP_EST_COUNT – sp_est_cnt + sp_max_cnt)
        sp_est_cnt  = 0
        sp_max_cnt = 0
        sp_max = 0
sp_est_cnt = sp_est_cnt + 1
if (in_level > MIN_SPEECH_LEVEL1) AND ((vadreg = 1) OR (in_level > speech_level))
        sp_max_cnt = sp_max_cnt + 1
        sp_max = MAX(sp_max, in_level)
        if (sp_max_cnt > SP_ACTIVITY_COUNT)
                if (sp_max > MIN_SPEECH_LEVEL2)
                    if (sp_max > speech_level)
                            speech_level = speech_level + SP_ALPHA_UP * (sp_max – speech_level)
                    else
                            speech_level = speech_level + SP_ALPHA_DOWN * (sp_max –
speech_level)
                sp_max_cnt = 0
                sp_max = 0
                sp_est_cnt = 0
```

4 Computational details

A low level description has been prepared in form of ANSI C source code [1].

Annex A (informative) :
Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New