IMPROVED PACKET LOSS RECOVERY USING LATE FRAMES FOR PREDICTION-BASED SPEECH CODERS

Philippe Gournay, François Rousseau, and Roch Lefebvre

Speech and Audio Research Group
Dept. of Electrical & Computer Engineering
University of Sherbrooke
Sherbrooke (Quebec) J1K 2R1 CANADA

VoiceAge Corporation
750 chemin Lucerne, Suite 250
Montreal (Québec) H3R 2H6 CANADA
E-mail: Philippe.Gournay@USherbrooke.ca

ABSTRACT

This paper presents a method to improve the recovery of a speech decoder after the reception of one or several late frames. Rather than considering a late frame as "lost", we propose to use it in order to update the internal state of the decoder. This limits, and in some cases stops, the error propagation caused by the concealment. Evaluation results show that there is a lot to be gained in a voice over IP environment, where late frames can be used to improve the robustness against jitter without increasing the overall end-to-end delay.

1. INTRODUCTION

Speech coding is a key technology for efficient voice communications over both wireless and wireline digital networks. Prediction-based speech coders are known to offer a very good compromise between bit rate and voice quality. CELP coders in particular now prevail in the 4 to 24 kbits/s range. However, these coders are known to be sensitive to bit errors and packet losses because of inter-frame dependencies in their predictor states.

In the specific context of voice over packet networks, one or several encoded speech frames are grouped into a single packet, which represents a data block on the network. The transit time for this packet through the network varies due to queuing effects along the transmission path. A "jitter" or "playout" buffer, that allows the receiver to wait for all packets arriving within an acceptable time limit, is used to control the effects of such variability. However, some packets may still arrive too late to be decoded. Missing or late packets are usually considered as "lost", and a concealment procedure has to be applied to replace the missing audio samples. Unfortunately, concealment is not perfect and errors introduced in the concealed frame propagate in the following ones.

In this paper, we will show that there is a lot to be gained by using late frames rather than considering them as lost. They can be used to improve the robustness of the decoder without increasing the overall end-to-end delay.

The paper is organized as follows. The sensitivity of prediction-based speech coders to frame losses, and the problem of error propagation after frame losses, are briefly discussed in section 2. Existing methods to improve the robustness of a speech decoder against frame losses are reviewed in section 3. The proposed method using late frames is described in section 4. Some evaluation results obtained with the AMR-WB speech coder [1] are given in section 5. Finally, conclusions are drawn in section 6.

2. SENSITIVITY TO FRAME LOSSES

The internal state of an encoder (and of the corresponding decoder) includes, in particular, the past samples required for long-term and short-term prediction, and a memory for predictive quantizers. When all frames are received correctly (i.e. no bit errors or lost packets) the encoder and decoder predictor states are identical. The speech signal generated by the decoder is then "correct", i.e. identical to the local synthesis at the encoder side. When one or more frames are lost, the decoder has to apply a concealment procedure in order to generate the missing audio samples. This procedure produces some distortion, even if in many instances the concealed speech retains much of the missing speech structure. Moreover, it does not update correctly the internal state of the decoder. Therefore, due to the highly predictive nature of modern coders, errors introduced in the concealed frame also propagate in the following ones even if the decoder receives the corresponding packets correctly.

More details on the sensitivity of CELP coders to frame losses can be found in [2]. In particular, this reference gives a detailed analysis of the impact of frame losses occurring at different moments on the recovery time for the ITU G.729 narrow band speech coder.

3. EXISTING METHODS TO IMPROVE THE ROBUSTNESS OF A DECODER

Following is a very brief review of the most common and effective methods to improve the robustness of a speech decoder against packet losses. A more thorough review can be found in [3].

The aim of this review is threefold. First, it shows that most methods rely either on a higher bit rate or a higher delay to improve the robustness. Then, it shows that, to date, much less work has been dedicated towards improving the recovery of a decoder after frame losses than towards improving the concealment itself. Finally, it underlines the fact that late packets are generally not taken into account, even though they are very frequent in a VoIP context.

3.1. Sender-based methods

Sender-based methods, such as forward error correction (FEC), essentially make use of redundancy. Best results are obtained when the amount of redundancy is varied according to speech [2] or channel [4] properties. Multiple description coding (MDC) is even more efficient, but it requires a set of independent transmission paths [5].

Retransmission-based methods form a subclass of sender-based methods that involve both the sender and the receiver. Packets are retransmitted only when needed. Those methods are rarely used in full-duplex communications because of the unacceptable additional transmission delay they require.

One method is worth mentioning nevertheless, because it relies on late packets. The RESCU (Recovery from Error Spread Using Continuous Update) method was originally proposed for video conferencing applications [6] but its extension to audio is somewhat straightforward. In this method, video frames are displayed at their normal playout time so that no additional delay is introduced. Whenever a reference frame is considered as lost, the receiver asks for its retransmission. If the retransmitted packet arrives too late to be displayed, concealment is applied. However, a late retransmitted packet can be used to restore the concealed reference frame, which stops error propagation among predicted frames.

3.2. Receiver -based methods

Most receiver-based methods rely on buffering. A playout delay that can be either fixed or adaptive [7] is introduced so that the reception buffer never empties. This lowers the number of lost packet but at the expense of an increase in end-to-end delay. Also, continuous adaptation of the playout delay requires some kind of speech rate adaptation that may degrade the speech quality.

Other reception-based methods rely on the intrinsic robustness of the decoder. Most speech decoders now include a frame loss concealment procedure that can be activated by setting a bad frame indicator (BFI). The decoder then uses the parameters of the previous frame to extrapolate those for the lost frame (see for example [1]). An improved recovery procedure may also be used to foster the resynchronization of the decoder. In the method described in [8], frames received immediately before and after the packet loss are used to update the internal state of the decoder. However, this method does not make use of late packets. Finally, we must mention the ILBC codec proposed by the IETF, since it was specifically designed to exhibit a short recovery time: the inter-frame dependency is minimized at the encoder side, but at the cost of a significant increase in bit rate [9].

4. UPDATING THE INTERNAL STATES USING LATE FRAMES

4.1. Description of the method

Fig. 1 illustrates the effects of one late frame in the original decoder (line B) and in the decoder with the update capability (line C). Correct output is shown in white. Error propagation is shown in gray. The output of the decoder without any lost or late frame (or equivalently, the output of the encoder's local decoder) is given on line A. This diagram serves as a visual aid to describe the processing that the update method applies.

Binary frames are received and decoded normally up to frame *n-1*. Frame *n* is not available in time for the decoding. The "concealment" procedure generates some replacement audio that differs form the expected audio. Since the internal state of the decoder is not updated correctly in the original decoder, the error introduced in frame *n* propagates in the following ones (line B).

Suppose now that frame *n* arrives at the decoder before the decoding of frame *n+1* (line C). There are two possible choices: (i) throw away the content of frame *n*, use the "bad" internal state produced by the concealment, and decode frame *n+1* as it is done in the original decoder; or (ii) restore the internal state of the decoder to its value at the end of frame *n-1*, decode frame *n* without outputting the decoded speech (which results in updating the internal state to its "good" value), and (iii) decode frame *n+1* as if no error had occurred.

In practice, some smoothing is required to prevent any discontinuity at the boundary between frame *n* and frame *n+1*. This can be done by weighting signals (i) and (iii) in Fig. 1 with fade-in, fade-out windows. Best results are obtained when this is done in the excitation domain. In that case, the memories of synthesis filters should be taken from the internal state following the concealment (actual past synthesized sampled).

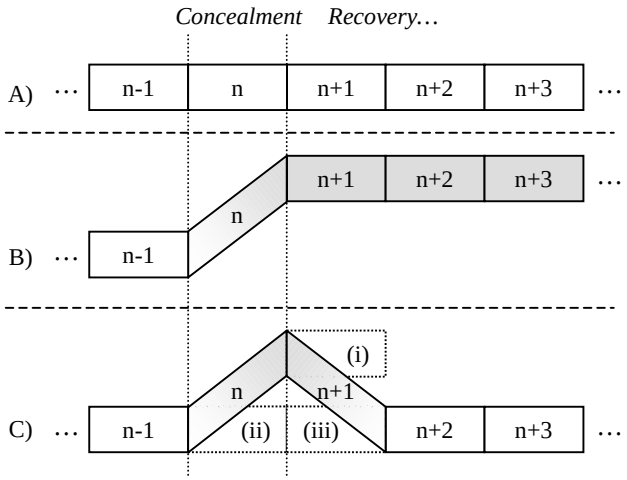

Fig. 1: *Chronogram showing the effects of one late frame.*

4.2. Sample signals

Fig. 2 shows some signal examples. Line 1 shows the decoded signal without any frame loss. Note that the original and modified decoders have exactly the same output when no update operation is performed. Line 2 shows the output of the original decoder when the third frame is lost. Since this loss occurs during a voiced onset, it triggers a strong energy loss (spanning one complete phoneme) and a high distortion level. In that case, the recovery time is long (line 4). Line 3 shows the output of the modified decoder when an update is performed after the concealment. Since all the necessary information was made available to the decoder in time to be taken into account, the recovery is fast and complete (line 5). All signals are represented at the same amplitude scale.

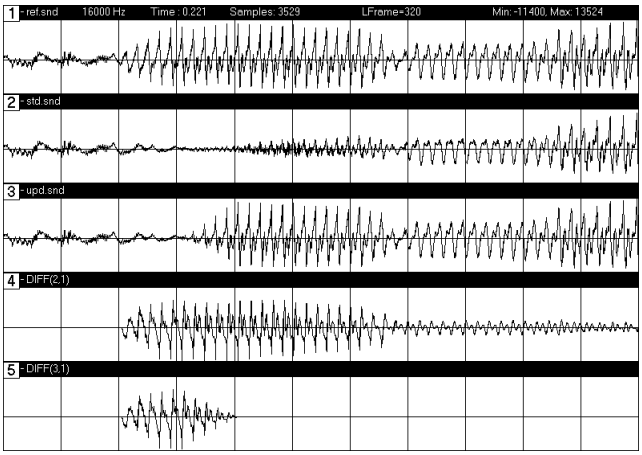

Fig. 2: *Decoded speech signals.*
1. no frame loss. - 2. original decoder, 3rd frame was lost.
3. modified decoder, 3rd frame was late. – 4. and 5. error signals for the original and modified decoders.

4.3. Complexity issues

The memory requirements for the update method consists of as many copies of the internal state structure as of frame we want to be able to go back in the past. If we limit the update capability to one single frame in the past (i.e. a frame won't be taken into account if it is delayed by more than one frame duration after its normal playout delay) then only one copy of the internal state is required.

In terms of processing power, if the update capability is limited as above and if the smoothing is performed in the excitation domain, then the complexity overhead is roughly equal to two decodings of the excitation (signals (i) and (ii) in Fig. 1). In the special case of the AMR-WB decoder this approximately doubles the processing power for frame *n+1* (including the update operation).

In that case again, the software interface for the modified decoder can be made very simple. The usual bad frame indicator (BFI) is used to activate the concealment procedure. An extra flag "UPD" is used to indicate that a frame is given for update only (no audio is generated). All the internal machinery is hidden to the user. The call sequence is as follows:

```
Decode(Bits_{n-1}, Audio_{n-1}, BFI=0, UPD=0)
Decode(  -  ,  Audio_n,  BFI=1, UPD=0) //Conceal
Decode(Bits_n,    -   ,  BFI=0, UPD=1) //Update
Decode(Bits_{n+1}, Audio_{n+1}, BFI=0, UPD=0) //Recover
Decode(Bits_{n+2}, Audio_{n+2}, BFI=0, UPD=0) // ...
```

Fig. 3: *Call sequence for the modified decoder*

5. EVALUATION RESULTS

The update capability described above was implemented in the AMR-WB decoder. A subjective evaluation was conducted in order to quantify the improvement obtained by this method.

5.1. The evaluation procedure

The evaluation was a pairwise comparison between the original AMR-WB decoder "STD" and the decoder with the update capability "UPD". The bit rate was fixed at 15.65 kbits/s and the DTX was disabled. We considered 3 packet loss conditions that are discussed below. Losses were synchronous on both decoders. The test material consisted of 50 sentences from the French corpus "BDSON" (4 speakers, 30 female and 20 male sentences). The average sample length was 2.5 seconds (120 frames). The listeners were presented twice with both coded version of a sentence (AB-AB, with A and B being either "STD" or "UPD"). They had 3 choices (A, B or indifferent) and no limit on the time to vote.

Since there were 3 error conditions, 50 sentences, and 2 possible presentation orders, 300 stimuli were generated. Each stimulus was evaluated once and only once by one of the 10 listeners. The sentences, speakers, conditions and presentation orders were randomized and balanced among the listeners.

The percentage of preference for each condition is given in Table 1. 100 votes were collected per condition. Line "prob" gives the probability that the observed preference is due only to random effects (sign test).

5.2. Condition 1: One late frame

In condition 1, the receiver was fed with one late frame every 10 frames. Late frames were considered as lost by the original decoder, while they were used for the update in the modified decoder (same call sequence as in Fig. 3). Therefore, there was no error propagation in the modified decoder after the end of the frame following the concealed one. Listeners' votes show a strong preference toward the modified decoder.

5.3. Condition 2: One lost frame and one late frame

In condition 2, one lost frame every 15 frames was immediately followed by one late frame. In that case, the update mechanism does not completely stop the error propagation. In nearly half of the cases, the listeners were unable to express any preference. However, the great majority of the expressed preferences went to the modified decoder, which is statistically highly significant.

5.4. Condition 3: Three consecutive late frames

In condition 3, the receiver was fed with 3 consecutive late frames every 20 frames. This scenario is representative of an adaptive jitter buffer that would fail to adapt completely (playout delay too short). In that case, the original decoder conceals 3 consecutive frames. This often results in a significant energy loss, a high distortion level, and a long recovery time. On the other hand, systematically updating the internal state after each concealed frame leads to very few distortions and no error propagation.

Since there is such a strong preference towards the modified decoder, the AB test may not have been the most appropriate one. However, it has the advantage of showing how systematic and significant the improvement is.

	Total	**Cond. 1**	**Cond. 2**	**Cond.3**
UPD	70%	67%	48%	94%
=	25%	29%	41%	4%
STD	6%	4%	11%	2%
Prob.		$\sim 5.10^{-14}$	$\sim 10^{-6}$	$< 10^{-15}$

Table 1: *Evaluation results.*

6. CONCLUSION

We presented a method to improve the recovery of a speech decoder after the reception of one or several late frames. Rather than considering a late frame as "lost", we proposed to use it in order to update the internal state of the decoder. This method was implemented in the AMR-WB coder, but it would also apply to any other prediction-based speech, audio or video coder.

Evaluation results show that there is a lot to be gained in a voice over IP environment by using late frames rather than considering them as lost. Making use of late frames increases the robustness of the decoder against unpredictable jitter variations, without increasing the overall end-to-end delay. Conversely, making use of late frames allows the receiver to operate with a shorter playout delay (which reduces the overall end-to-end delay) without overly degrading the speech quality.

7. REFERENCES

[1] ETSI 3GPP TS 26.191, "AMR Wideband Speech Codec; Error concealment of erroneous or lost frames", March 2001. A detailed algorithmic description of the AMR-WB codec can be found in other 3GPP documents.

[2] H. Sanneck and N. T. L. Le, "Speech property-based FEC for Internet telephony applications", SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000), San Jose, CA, January 2000.

[3] B.W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmission over the Internet", IEEE International Symposium on Multimedia Software Engineering, December 2000.

[4] C. Padhye, K. Christensen, and W. Moreno, "A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications", IEEE International Performance, Computing and Communication Conference, February 2000.

[5] W. Jiang and A. Ortega. "Multiple description speech coding for robust communication over lossy packet networks", International Conference on Multimedia and Expo, volume 1, pp. 444-7, Aug. 2000. New York, NY, USA.

[6] I. Rhee, "Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet", *Computer Communication Review*, ACM SIGCOMM, vol. 28 n. 4, pp. 290–301, October 1998.

[7] Y.J. Liang, N. Farber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications", Proc. of the ICASSP, vol. 3, pp. 1445-1448, May 2001.

[8] N. Naka and T. Ohya (NTT Mobile communications), "Updating internal states of a speech decoder after errors have occurred", U.S. Patent US006085158A, 4 July 2000.

[9] S.V. Andersen, et al., "ILBC - A linear predictive coder with robustness to packet losses", IEEE Workshop on speech coding, pp. 23-25, October 2002.