Removing Long Echo Delay Using Combination of Jitter Buffer and Adaptive Filter

Dinh Van Phong, Nguyen The Hieu,
Nguyen Huy Tinh, Dinh Viet Quan
Viettel Network Technologies Center, Viettel Group
Email: phongdv6@viettel.com.vn

Tran Duc Tan
University of Engineering & Technology
Vietnam National University, Hanoi
Email: tantd@vnu.edu.vn

***Abstract*— Echo in telephone transmission systems is a serious problem. It affects and distorts the desired speech. Echo cancellation methods have studied for a few decades and standardized in ITU G.168. The core theory in echo cancellation methods is using an adaptive filter which uses one of these algorithms: LMS (least mean square), NLMS (normalized least mean square), RLS (Recursive least squares), etc to remove the echo. In fixed conditions, these algorithms are efficient to remove echo. However, in some real telecom environments, with long echo delays, we must increase the filter length to a big value. But, it's not efficient in performance due to high computational complexity. In this paper, we propose a solution that uses a jitter buffer along with an adaptive filter to compensate long echo delays. This solution demonstrated its efficiency in Viettel Network - the biggest telecom provider in Vietnam - both on voice quality and system performance.**

Keywords*— *Acoustic Echo, Line Echo, Jitter Buffer, Adaptive Filter.

I. INTRODUCTION

A. Echo types in telecommunication systems

Echo is defined as a delayed and distorted version of an original sound or signal which is generated when it is reflected back to the source [1]. There are two echo types. The first echo type called *line echo* is generated in the hybrid transformer when two-wire to four-wire conversion in Public Switched Telephone Network (PSTN) network is used. This type is illustrated in Fig 1.

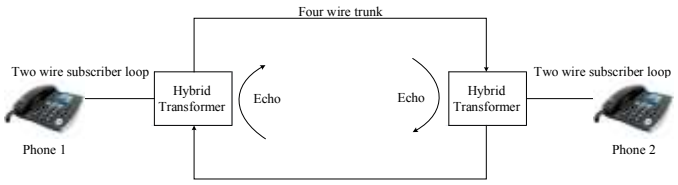

Fig. 1. Line Echo

The second type called acoustic echo is generated by reflecting voice signals between microphone and loudspeaker of a handset. This type is illustrated in Fig 2.

In real telecom environments, acoustic echo is processed so well on a mobile handset, nobody complains about acoustic echo on his/her phone. However, we still meet line echo when making a call from a 3G mobile to a PSTN subscriber.

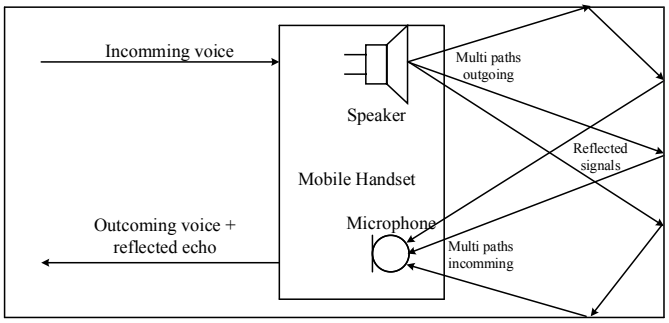

Fig. 2. Acoustic Echo

B. Background of echo cancellation

Echo cancellation methods have been studied for few decades. It began in Bell Lab in 1962 [2], and published by Mr. Sondhi with a series of paper [3][4][5][6]. The main idea of echo cancellation is to generate a synthetic replica of the echo by feeding the far end signal into an adaptive filter and to subtract it from the return signal [2]. The concept "adaptive filter" means that the filter automatically drives itself to match its characteristic to whatever echo path. Figure 3 illustrates an adaptive filter used for echo cancellation.

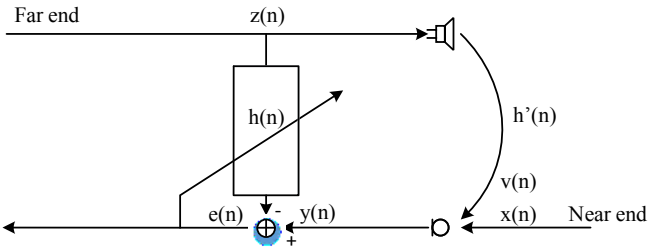

Fig. 3. Echo cancellation using an adaptive filter

In an ideal model, we assume that there is no noise. So, the near end signal *y(n)* is the combination of the cleared near end signal *x(n)* and the echo signal *v(n)*.

$$y(n) = x(n) + v(n) = x(n) + z(n)*h'(n), \quad (1)$$

Where *h'(n)* is the environment impulse response. We expect to minimize *v(n)* to be zero. To do that, the far end signal *z(n)* is passed to an FIR filter *h(n)*, its output is subtracted by *y(n)*.

$$e(n)= y(n) – z(n)*h(n) = x(n) + z(n)*h'(n) – z(n)*h(n) \quad (2)$$

We expect an ideal result, $e(n) = x(n)$, it means that $z(n)*h'(n) – z(n)*h(n) = 0$. In that case, we say that $h(n)$ converged to $h'(n)$.

The simplest algorithm used to make $h(n)$ converge to $h'(n)$ is least mean square (LMS). Many authors later focused to optimize algorithm rate and developed other versions of this algorithm such as normalized least mean square (NLMS), proportionate Normalized Least Mean Squares (PNLMS) algorithm [7], robust variable step-size NLMS (RVSS-NLMS) algorithm [8], recursive least square (RLS) [11] and recent researches [9][10]. Among these algorithms, NLMS is often used since it is simple in implementation and standardized in ITU-T G.168 [12].

Table 1. NLMS algorithm summary

Parameter	L = filter length μ = step size
Initialization	h(0) = zeros(L)
Computation	For n = 0, 1, 2…. $\mathbf{z}(n) = [z(n), z(n-1),\ldots,z(n-L+1)]^T$ $e(n) = y(n) – \mathbf{h}^H(n)\,\mathbf{z}(n)$ $\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu e^*(n)\mathbf{z}(n)}{\mathbf{z}^H(n)\mathbf{z}(n)}$

The FIR filter length related to the tail length (i.e. the delay) can be processed by NLMS. For example: 256 (32 ms), 512 (64 ms), 1024 (128 ms)… It can be deployed as software [13][14] or hardware [15].

C. Limitations

- Higher filter length, larger computation**:** Table 1 showed that, for each step, the algorithm needs L multiplications and L+1 additions for e(n) computation and 2×L multiplications and L additions for filter coefficient updating. For an input voice signal sampled at 8000 Hz, the number of multiplication can be calculated:

$$M = 3L \times 8000 \text{ Multiplications/s,} \quad (3)$$

and the number of additions can be calculated:

$$A = 2L \times 8000 \text{ Additions/s.} \quad (4)$$

For example, assuming a filter length of 512, the number of multiplication/s is ~12.2 × 106 and the number of addition/s is ~8.2 × 106.

In a voice processing system, to ensure the real-time constraints, M and A need to be decreased. It means that the lower the filter length, the better the system performance. But the lower in filter length, the lower tail length can be processed. This is one of the most difficulties in implementing NLMS algorithm.

- How to process long echo delay: In a real telecom environment - Viettel Network - we measured the echo delay about 350 ms, it means that we need a filter length of about 2800. This exceeds availability of some companies [13][14] due to the number of computations in NLMS algorithm is so high.

Two above limitations are difficulties that excite our engineers to find a solution to solve them. Our idea is to find a solution to decrease the difference delay between the far end $z(n)$ and the echo signal $v(n)$ before they are fed into the adaptive filter. It is impossible to decrease $v(n)$ delay because it is the transmission line delay, it can be assumed as a constant. But we can delay the far end $z(n)$, therefore the difference delay between $z(n)$ and $v(n)$ will decrease.

II. Proposal Of Using Jitter Buffer In Compensating Echo Delay

A. Processing In Internet Protocol (IP) Environment

Fig. 3 and the above overview are suggested in the consecutive time domain. In which, the far end signal $z(n)$ and the echo signal $v(n)$ are consecutively transmitted by time division multiplexing (TDM) technique. But, nowadays, most modern telecom systems are running based upon IP platform. It means that the voice data are framed and sent as discrete packets. The voice data packet parameters depend on the codec types used, table 2 describes some voice data parameters of some audio codecs.

Table 2. Some codec types and packet size

No	**Codec type**	**Packet size** (byte)	**Duration** (ms)
01	G.711 (PCMA/PCMU) [22]	80	10
02	AMR narrow band [23]	31 (rate 12.2 kbps)	20
03	AMR wide band [24]	62 (rate 23.85 kbps)	20
03	GSMFR [25]	33	20
04	GSMHR [26]	14	20

Thus, both $z(n)$ and $v(n)$ are in packet format. To delay $z(n)$, we can find a solution to delay its packets.

B. Jitter Buffer

Jitter Buffer is a concept in computer network [16]. It is a queue running based upon first in first out (FIFO) law. It stores voice data packets. If a jitter buffer has a size of N, it

means that a packet will be stored in the jitter buffer with the time of N × packet duration.

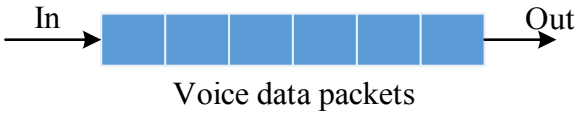

Fig. 4. Illustration of a Jitter Buffer

For example, the PCM packets are transmitted at 10ms of duration. They are fed into a jitter buffer which has 10 elements, so the total delay time of a packet in the jitter buffer is 100ms.

C. *Proposal Model Of Using Jitter Buffer In Compensating Echo Delay*

The jitter buffer is used to store the input voice packets from the far end before feeding into the adaptive filter (see Fig. 5).

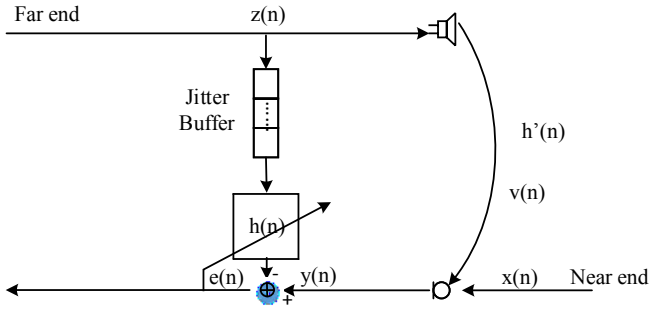

Fig. 5. Proposed model of using a jitter buffer in compensating echo delay.

Assuming that a packet (T *ms* of duration) goes outside the 3G system at t_0 , and the echo comes back to the system at t_1 , the filter length of L , the echo delay can be calculated:

$$D = t_1 - t_0 \tag{5}$$

So, the jitter buffer size S can be calculated:

$$S = \frac{\left(\frac{D \times 8000}{10^3}\right) - L}{\left(\frac{T \times 8000}{10^3}\right)} \tag{6}$$

III. RESULTS AND DISCUSSIONS

A. *Measurement Methods*

The voice signals used in our test captured from real transmission lines in Viettel Network (see Fig. 6), the echo delay ~350 ms is measured in the time domain by some audio analyzers such as Audacity, Sonic Visualiser. The signals in our tests are noted as following:

- ***Cleared original voice signal (org_sig):*** The cleared voice signal with no echo is sent from 3G Mobile Network.
- ***Original echo voice signals (org_echo_sig):*** The signal with echo is captured in real PSTN line Viettel Network.
- ***Cleared voice signals by NLMS (clr_nlms_sig):*** The cleared voice signal after using NLMS on the original echo voice signal.
- ***Cleared voice signals by NLMS combined with jitter buffer (clr_nlms_jitter_sig):*** The cleared voice signal after using NLMS combined with a jitter buffer on the original echo voice signal.

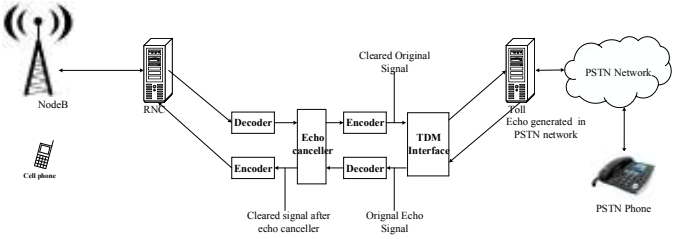

Fig. 6. The testing model in Viettel network

The captured signals are compared in 03 ways:

- **In the time domain:** This is the basic way to see how echo signals are cancelled when using NLMS and NLMS combined with the jitter buffer. The signals are compared by their amplitudes.
- **Signal To Noise Ratio (SNR)**: Both ***clr_nlms_sig*** and ***clr_nlms_jitter_sig*** are compared with ***org_sig.***
- **Mean Opinion Score (MOS):** MOS is defined as an international standard in ITU P.863 "Perceptual objective listening quality prediction" [17]. There are some software tools which are compatible with this standard such as: VQT from GL [18], Opera Voice/Audio Quality Analyzer from Opticom [19]. In our lab, we use VQT from GL to score the testing signals.

B. *Results*

In the below figures, we compare the results between using jitter buffer and not using jitter buffer.

Fig. 7 displays voice signals in time domain, L = 1024, D = 350 ms. In which, all signals are compared by its amplitude. We can observe that by using jitter buffer, we can get the lower amplitude of the echo remaining in the original signal.

Fig. 7. (1) org_sig, (2) clr_nlms_sig, (3) clr_nlms_jitter_sig

In Fig. 8, the signal to noise ratio (SNR) is used to compare the voice signals. SNRs are measured by using VQT software

at many filter lengths. We can observe that by using jitter buffer, we can get the better SNR in the output signals.

In the table 3, we use VQT software to score voice signals by MOS. The MOS are divided into 05 levels [18]: *disregard, poor, fair, good and excellent*. We can observe that in the case of without jitter buffer, we must use the filter length L = 1024 to get the "good" score, but this result can be got by using the filter length L = 256 in the case of using jitter buffer. It means that we decreased 04 times in algorithmic complexity to have the same result. In another viewpoint, if we use the filter length L = 1024 in both cases, we get "good" score without jitter buffer, but we can get "excellent" score in the case of using jitter buffer.

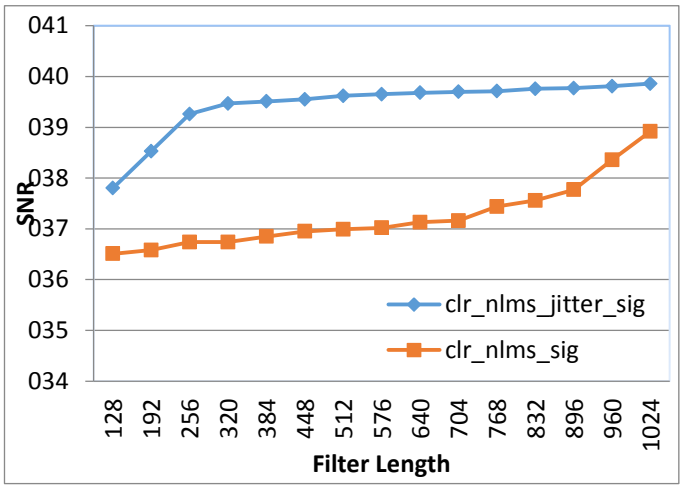

Fig. 8. SNR compared between clr_nlms_sig and clr_nlms_jitter_sig

Table 3. MOS score comparison between *clr_nlms_sig* and *clr_nlms_jitter_sig*

Filter Length	Clr_nlms_sig	Clr_nlms_jitter_sig
128	Fair	Fair
256	*Fair*	*Good*
320	Fair	Good
384	Fair	Good
448	Fair	Good
512	Fair	Good
576	Fair	Good
640	Fair	Good
704	Fair	Good
768	Fair	Good
832	Fair	Good
896	Fair	Good
960	Fair	Good
1024	*Good*	*Excellent*

IV. CONCLUSION

In this paper, we have succeeded in processing the long echo delays in Viettel network by using a combination of jitter buffer and adaptive filter. Experiment results shown that using a jitter buffer to compensate echo delay is extremely efficient. We applied NLMS (L = 256) combined with a jitter buffer in Viettel Network and got better qualities both on voice quality and system performance. In future work, this buffer will be designed to embed into the adaptive filter. Also, the results of this study can be combined with the coding techniques for the specific applications [20][21].

ACKNOWLEDGMENT

This paper is one of results in the project "Researching & Developing Gate Mobile Switching Center, code: 002-18-TĐ-RĐP-DS", sponsored by Viettel Network Technologies Center, Viettel Group.

REFERENCES

[1] Kazuo Murano, Shigeyuki Unagami, Fumio Amano, "Echo Cancellation and Applications", IEEE Communications Magazine, 49 – 55 (January, 1990).
[2] M. M. Sondhi, "The history of echo cancellation," IEEE Signal Process Mag., Vol. 23, No.5, Sep. 2006, 95-98.
[3] M. M. Sondhi andA. J. Presti, 'A Self-AdaptiveEcho Canceler,'BSTJ, vol. 45, p. 1,851, 1966.
[4] M.M. Sondhi, "An adaptive echo canceler," *Bell Syst. Tech. J.*, vol. 46, no. 3, pp. 497–511, Mar. 1967.
[5] M.M. Sondhi, "Closed loop adaptive echo canceller using generalized filter networks," U.S.Patent 3 499 999, 1970.
[6] M.M. Sondhi and D. Berkley, "Silencing echoes on the telephone network," *Proc. IEEE*, vol. 68, no. 8, pp. 948–963, 1980.
[7] Donald L. Duttweiler, "Proportionate Normalized Least-Mean-Squares Adaptation in Echo Cancelers", IEEE Trans. Audio Speech, pp. 508 – 518, 2000.
[8] Insun Song, Won Il Lee, Nam Kyu Kwon, and PooGyeon Park, "A Robust Variable Step-Size NLMS Algorithm Through A Combination of Robust Cost Functions", International Journal of Information and Electronics Engineering, Vol. 2, No. 6, November 2012.
[9] Jean-Marc Valin, "On Adjusting the Learning Rate in Frequency Domain Echo Cancellation With Double-Talk", IEEE Trans. Audio Speech, pp. 1030 - 1034, 2007.
[10] Jean-Marc Valin, Iain B. Collings, "a new robust frequency domain echo canceller with closed-loop learning rate adaptation", ICASSP, pp. 93 – 96, 2007.
[11] Simon Haykin: Adaptive Filter Theory, Prentice Hall, 2002, ISBN 0-13-048434-2.
[12] ITU-T G.168, "Digital network echo canceller", April 2015.
[13] G.168 Echo cancellation line, network & packet, http://www.adaptivedigital.com/vqe-suite/g-168/, access: June 26, 2018.

[14] Line/Network echo canceller, https://www.vocal.com/echo-cancellation/line-network-echo-canceller/, access: June 26, 2018.
[15] Mahmod. A. Al Zubaidy, "Hardware Implementation for the Echo Canceller System based Subband Technique using TMS320C6713 DSP Kit", International Journal of AdvancedComputer Science and Applications, Vol. 9, No. 1, 2018.
[16] Comer, Douglas E. (2008). Computer Networks and Internets, Prentice Hall. p. 476, ISBN 978-0-13-606127-4.
[17] ITU P.863 "Perceptual objective listening quality prediction". March 2018.
[18] Voice Quality Testing (VQT) Software (POLQA, PESQ), https://www.gl.com/voice-quality-testing-pesq-polqa.html, access: June 26, 2018.
[19] Opera Voice Quality Analysis, http://www.opticom.de/products/opera.html, access: June 26, 2018.
[20] Tam Vu Van,Tran Duc-Tan, Phan Trong Hanh (2017). Data embedding in audio signal using multiple bit marking layers method. Multimedia Tools and Applications, 76(9), 11391-11406.
[21] Vu, V. T., Tran, D. T., Nguyen, D. T., Nguyen, T. T., & Phan, T. H. (2015). Data embedding in audio signal by a novel bit marking method. International Journal of Advancements in Computing Technology, 7(1), pp. 67-76.
[22] ITU G.711: Pulse code modulation (PCM) of voice frequencies; ITU-T Recommendation (11/1988), Retrieved on 2009-07-08.
[23] 3GPP TS 26.090 - Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions". 3GPP. Retrieved 2010-07-21.
[24] ITU-T (2003) ITU-T Recommendation G.722.2 Page i. Retrieved on 2009-06-17.
[25] ETSI EN 300 961 V8.1.1 (2000-11) - (GSM 06.10 version 8.1.1 Release 1999), Retrieved on 2009-07-08.
[26] ETSI, EN 300 969 - Half rate speech transcoding (GSM 06.20 version 8.0.1 Release 1999), Retrieved on 2009-07-11.