Adaptive Playout Control and Signal Reconstruction for Speech-Based Audio Convergence VoIP

Jun-Yong Lee
Department of Electronics Convergence Engineering
Kwangwoon University
Seoul, South-Korea
Jasonlee88@kw.ac.kr

Hyoung-Gook Kim
Department of Electronics Convergence Engineering
Kwangwoon University
Seoul, South-Korea
hkim@kw.ac.kr

***Abstract*—This paper proposes an adaptive playout control and signal reconstruction method for speech-based audio convergence VoIP. In adaptive playout control, the buffering time is minimized by way of playing out normally or compressing each packet according to accurate network jitter estimation. Also, linear prediction-based signal reconstruction recovers lost packets and minimizes boundary discontinuities between the good packets and the reconstructed packets. The proposed receiver-based enhancing method delivers high-quality voice and music service over IP networks.**

Keywords-playout control; signal reconstrution; jitter estimation

I. INTRODUCTION

The use of Voice over Internet Protocol (VoIP) for carrying real-time voice data over any IP network has significant impacts on the telecommunication industry.

However, a number of factors may affect the service quality of VoIP, such as packet loss, packet delay, and network delay variation (also known as "jitter"). To provide reliable services with satisfactory voice quality over IP networks, considerable efforts [1][2][3][4][5] have been made within different layers of current communication systems to reduce delay, smooth jitter, and recover loss.

Some techniques have been developed for concealing packet loss. In waveform substitution method, the missing frames are replaced by another already-received frame using pitch replication [4], [6] or pattern matching [7]. And the model of the previously received signal (eventually slightly modified) is used to generate the missing signal [8] in model extrapolation method.

Several VoIP playout buffer scheduling or timing recovery algorithms have been proposed. Pinto [9] presented a method that adjusts silence periods between signal spurts to improve voice interaction quality, while Liang [3] proposed adaptive playout-buffer schedulers that adjust the voice regions by introducing time-scale modification. Chi [10], Li [5], and Aragao [11] suggest a playout scheduling method based on modeling packet arrival times using K-Erlang distribution, a Gaussian model, Pareto distribution, etc. However, these methods are "packet-based" and decide whether to stretch or compress a packet once it is received. Florncio [12] used the "buffer-based" method, which decides whether to stretch or compress only when the audio playout device needs a frame.

According to enhancing VoIP speech quality, more and more smartphone users are taking advantage of mobile VoIP services. Recently, speech-based audio convergence VoIP codecs [13] that offer a high or nearly transparent quality while remaining compliant with tight conversational requirements (delay constraints in two-way communication) are recently emerging for the applications of the high-quality conferencing and VoIP telephony. Specially, ITU-T Recommendation G.729.1 is a scalable wideband speech and audio coding standard designed to facilitate a graceful and cost-effective evolution to high-quality wideband speech based audio communications in packet-switched networks.

In this paper, we focus on enhancing VoIP speech and music quality only at receiving portion of a mobile Internet phone. The important functionality to be implemented at the receiver is an adaptive playout control and signal reconstruction scheme consisting of concealment of lost packets based on the redundancy in neighboring packets, adaptive playout-buffer scheduling using active jitter estimation, and smooth interpolation between two signals in a transition region.

Our method has three important improvements: 1) using accurate jitter estimation our playout-buffer control makes it possible to trade-off the buffering time with the rate of packet loss; 2) our signal reconstruction based on recursive linear prediction analysis and synthesis (LPAS) alleviates the metallic artifacts that are often introduced during concealing packet loss; and 3) using linear prediction (LP) based smooth interpolation between the two signals in a transition region, we improve VoIP speech and voice quality at the receiver.

This paper is organized as follows. Section II describes our proposed method. Section III discusses the experimental results. Finally, section IV presents our conclusion.

II. PROPOSED RECEIVER-BASED VOIP QUALITY ENHANCING METHOD

A. *Structure of the Receiving Part in VoIP System*

The proposed structure of the receiving part of a mobile Internet phone is illustrated in Fig. 1. The receiving system employs combined signal reconstruction and playout control (SRPC) on the decoded signal frames.

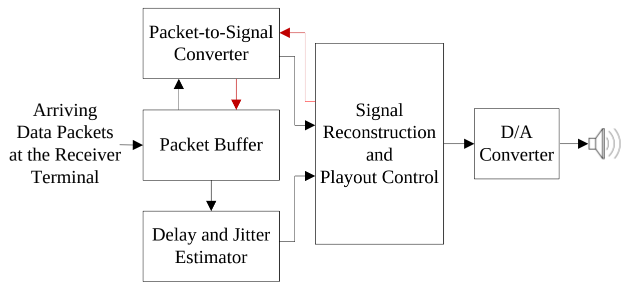

Figure 1. Structure of the receiving part of a mobile Internet phone.

Arriving voice data packets from the sender over IP network is adequately placed in a packet buffer. To feed arriving packets to packet-to-signal converter at regular intervals, the receiving system needs to maintain a packet buffer. In response to the arriving packets in the packet buffer, the network jitter is adaptively estimated and used to assign each packet a controlled playout time in the SRPC module. The packet buffer holds incoming packets and then releases them for decoding at a regulated speed (i.e., every 10 ms), thereby reducing system delay. In this paper, the length of each packet is 20 ms, and the size of the packet buffer as a storage medium is 200 ms. Therefore, 1–10 packets are present in the packet buffer.

The decoded signal frames are entered into the SRPC module. The current used packet-to-signal converter is G.729.1 decoder.

In the SRPC, one of three processing modes (loss concealment, smoothing, or timing recovery) is performed for recovering lost packets and controlling the playout time on the decoded signal frames.

To recover lost packets, the SRPC module often makes a subsequent frame demand from the packet-to-signal converter, causing the packet-to-signal converter to make a packet demand from the packet buffer. The packet buffer then extracts a voice or music data packet and sends it to the packet-to-signal converter, which decodes it as a signal frame. The digital to analog converter (D/A) regularly converts the sampled signal frame from SRPC into an analog signal. Finally, the user hears the analog voice or music signal through a speaker.

B. *Adaptive Playout Control and Signal Reconstruction*

Fig. 2 presents an overall flow chart of three processing modes in the SRPC shown in Fig. 1.

After silence segments are discriminated between signal frames coming from the signal frame buffer, the SRPC performs one of three modes on the i-th signal frame (where $i = 1, 2\ldots I$ denotes the time index when each voice packet is generated at the sending host or when each voice packet is played out through the speaker) and k-th arriving packet (where $k = 1, 2\ldots K$ presents the number of packet at the receiving host) as follows:

1) Loss concealment mode

If the i-th signal frame (subsequent signal frame for playing out) is absent from the signal frame buffer, a packet is declared lost and the "loss concealment mode" is entered.

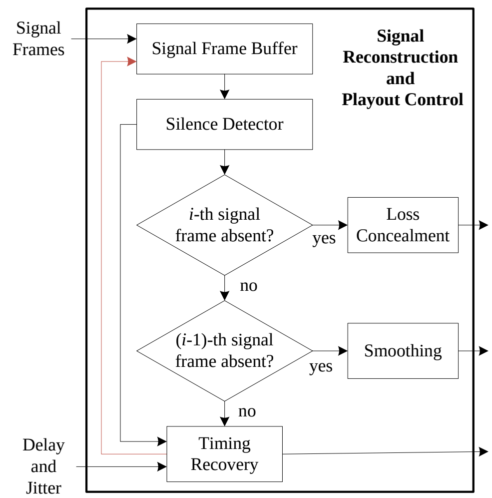

Figure 2. Flow chart of signal reconstruction and playout control.

Fig. 2 presents a flow chart of signal reconstruction and playout control. Our loss concealment algorithm is based on recursive LPAS using soft estimated pitch period to improve the G.722 Appendix IV PLC algorithm [5]. In the proposed recursive LPAS, the soft estimated pitch period is used to generate a smooth excitation for recovery of the lost packets, and effectively reduces tonal artificial frequencies that could be caused by repeating a small segment many times. If consecutive packets are lost, then the previous synthesized signal is recursively input to LP filter to generate the new smooth excitation signal. The reconstructed signal is synthesized by filtering the smooth excitation signal and gradually muted for the duration of the loss period.

2) Merging and smoothing mode

If the i-th signal frame is present in the signal frame buffer and the $(i$-1$)$-th signal frame was lost, discontinuity between the i-th signal frame and the $(i$-1$)$-th substituted signal frame occurs and the "smoothing mode" is entered. The smooth interpolation is obtained as follows: First, N samples from the i-th signal frame are obtained and input to the LPAS to generate the reference segment C. Second, the signal segment A most similar to the reference segment C is found in the samples of the $(i$-1$)$-th previous signal frame in the history buffer. Third, the smooth estimated signal frame D is generated using peak alignment overlap-add between the signal frame A and the signal frame C. Fourth, the $(i$-1$)$-th signal frame, the segment D, and the $(i$+1$)$-th signal frame are merged into the new segment O. The segment O is substituted into the i-th signal frame.

3) Timing recovery mode

If the i-th signal frame is present in the signal frame buffer and the $(i$-1$)$-th signal frame was not lost, the "timing recovery mode" is entered.

Fig. 3 depicts the algorithm flow chart of the decision of compression or normal playout process.

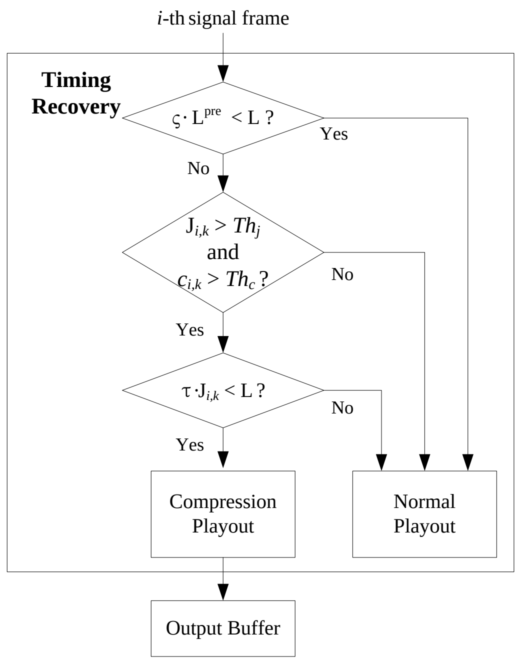

Figure 3. Flow chart of timing recovery.

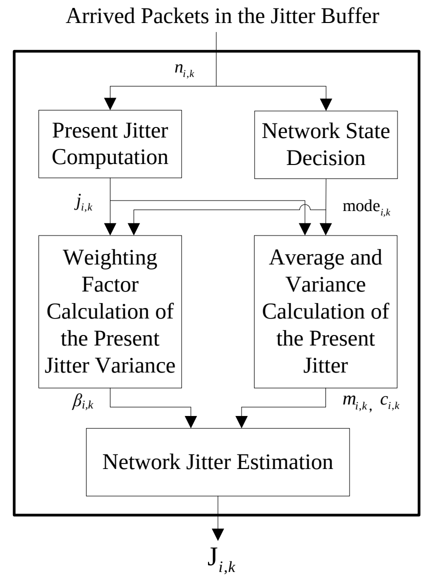

Figure 4. Block diagram of the proposed network jitter estimation.

The decision logic between normal playout or compression processes for the playout scheduling is performed using the estimated network jitter and the length of the remaining signal frames in the signal frame buffer. For this, the following subprocesses are handled:

- Let L and L^{pre} denote the total length of the current remaining signal frames and the previous remaining signal frames in the signal frame buffer, respectively. If L is larger than $\varsigma\cdot L^{pre}$ ($1 \leq \varsigma \leq 4$), the normal subprocess is initiated.
- If L is smaller than $\varsigma\cdot L^{pre}$, the estimated jitter $J_{i,k}$ is smaller than the jitter threshold Th_j ($10 \leq Th_j \leq 20$), and the jitter variance $c_{i,k}$ is smaller than the compression threshold Th_c ($3 \leq Th_c \leq 7$), then the normal subprocess is initiated.
- If the following three conditions are satisfied: $L < \varsigma\cdot L^{pre}$; $J_{i,k} > Th_j$; and $c_{i,k} > Th_c$; and $\tau\cdot J_{i,k} \geq L$ using τ($2 \leq \tau \leq 5$), then the normal subprocess is initiated.
- If the following four conditions are satisfied: $L < \varsigma\cdot L^{pre}$; $J_{i,k} > Th_j$; and $c_{i,k} > Th_c$; $\tau\cdot J_{i,k} < L$, then the compression subprocess is initiated.

C. *Network Jitter Estimation*

Our network jitter estimation incorporates spike detection and accurately predicts network delays including spike delays; thus, it is well-suited for timing recovery in playout algorithms in which playout delay is adjusted for each individual packet.

Fig. 4 depicts an algorithm flow chart of the proposed active jitter estimation.

The active jitter estimation is composed of five modules: present jitter computation, network state decision, weighting factor calculation of the present jitter variance, average and variance calculation of the present jitter variance, and network jitter estimation.

The network jitter estimation procedure is as follows:

(Step1) Inter-arrival jitter results in a network delay change. Therefore, the inter-arrival jitter of the *k*-th arriving packet in the *i*-th signal interval is computed.

(Step2) Using a modified enhanced normalized least mean squares algorithm (E-NLMS) [1], network state is classified into one of two zones: "*spike*" or "*normal*." The modified E-NLMS algorithm is incorporated with delay spike detection using weighting factor $\beta_{i,k}$ of the network jitter variance. A delay spike is detected when the actual network delay exceeds the predicted delay value or the previous delay by a threshold. When the delays drop down to the level before the mode is in force, the normal mode is switched.

(Step3) The weighting factor $\beta_{i,k}$ of the inter-arrival jitter variance can be obtained as:

$$\text{If } (\text{mode}_{i,k} = normal)$$

$$\beta_{i,k} = \begin{cases} \beta_{i,k-1} - \alpha_c, \; where \; \beta_{i,opt} < \beta_{i,k-1} \\ \beta_{i,k-1} + \alpha_d, \quad otherwise \end{cases} \tag{1}$$

else

$$\beta_{i,k} = \beta_{i,k-1}$$

using α_c ($0 < \alpha_c < 1$), α_d ($0.5 < \alpha_d < 1.5$) and

$$\beta_{i,opt} = \frac{j_{i,k} - m_{i,k-1}}{c_{i,k-1}} \tag{2}$$

 ISBN: 978-1-61208-342-1

where $\beta_{i,opt}$ is a weighting factor of optimal network jitter variance to minimize the jitter error incurred by varying network conditions.

(Step4) After the adjustments of $\beta_{i,k}$, average $c_{i,k}$, and variance $m_{i,k}$ of the inter-arrival jitter are calculated according to the determined network situation, as shown in (3):

$$\begin{aligned}
&\text{If } (\text{mode}_{i,k} = normal \text{ and } \text{mode}_{i,k-1} = spike) \\
&\quad m_{i,k} = \alpha \cdot m_{i,tmp} + (1-\alpha) \cdot j_{i,k}, \\
&\quad c_{i,k} = \alpha \cdot c_{i,tmp} + (1-\alpha) \cdot \left| m_{i,k} - j_{i,k} \right| \\
&\text{else} \\
&\quad m_{i,k} = \alpha \cdot m_{i,k-1} + (1-\alpha) \cdot j_{i,k} \\
&\quad c_{i,k} = \alpha \cdot c_{i,k-1} + (1-\alpha) \cdot \left| m_{i,k} - j_{i,k} \right|
\end{aligned} \tag{3}$$

where $\text{mode}_{i,k}$ and $\text{mode}_{i,k-1}$ represents the current and previous network state mode, respectively; α $(0 < \alpha < 1)$ is a smoothing parameter; *tmp* is the temporal point when the spike is detected; and $m_{i,tmp}$ and $c_{i,tmp}$ are the mean and the variance of the jitter at the point at which the previous spike was detected, respectively.

(Step 5) The active network jitter of the *k*–th arriving packet in the *i*–th signal interval is estimated using the calculated means and variances of the inter-arrival jitters as:

$$\mathrm{J}_{i,k} = m_{i,k} + \beta_{i,k} \cdot c_{i,k} \tag{4}$$

The estimate for the network delay $En_{i,k}$ is computed as

$$En_{i,k} = \alpha_n \cdot En_{i,k-1} + (1-\alpha_n) \cdot n_{i,k} \tag{5}$$

where α_n $(0 < \alpha_n < 1)$ is a weighting factor that controls the algorithm convergence rate and $n_{i,k}$ is the network delay that the *k*-th transmitted packet experiences.

The playout times are then adjusted as

$$p_{i+1,k} = En_{i,k} + R_{i,k} + \tau \cdot \mathrm{J}_{i,k} \tag{6}$$

where $R_{i,k}$ is the timing recovery delay, and τ $(0 < \tau < 3)$ controls the additional buffering delay and lateness loss ratio.

III. EXPERIMENTAL RESULTS

A. Testbed Infrastructure and Measurements

In order to evaluate the proposed adaptive playout control and signal reconstruction, a test bed is set up [14]. These are connected to each other by two types of networks: ethernet (100 Mbps) and wireless local area network (WLAN) (300 Mbps). SIP signaling involves a SIP Proxy Server and two *clients*. VoIP application is developed by using Visual C++ and is installed in *clients*. *Clients* are mobile devices that have the following specifications: 800 MHz CPU, and 4 GB memory. SIP signaling messages are transferred through the audio data transport module and are sent from clients to the SIP Proxy Server, and then the SIP signaling messages redirected to clients accordingly. Clients send audio data in RTP packets. In the audio data transport module, Clients A and B are each connected to access points (access points 1 and 2) of ipTime N604M (Hubs 1 and 2). In the network traffic emulator module, a traffic generator is used in order to simulate WLAN connections with different traffic loads such as delay, jitter, and packet loss.

The speech samples are digitized at 16 kHz. Each trace lasts for about 5 min and consists of 15,000 packets, each of which consists of 20 ms of speech content. The music samples consists of a database of 100 songs from different genres such as pop, hip-hop, jazz, and classical and are digitized at 16 kHz.

To evaluate the quality degradation, objective voice quality testing is performed using perceptual evaluation of speech quality (PESQ), total buffering delay (TBE), and jitter estimation error (JER). PESQ is a recognized method for accurately testing the quality level that will be perceived by a VoIP network user and is described in the latest ITU-T recommendation P.862 Amendment 2 [15]. PESQ provides a score ranging from 1 to 5, where 1 is unacceptable and 5 is excellent. A typical range for VoIP is 3.5 to 4.2. TBE and JER are defined as follows:

$$\mathrm{TBE} = \sum_{k=1}^{\mathrm{K}} \left(p_{i,k} - a_{i,k} \right) / \mathrm{K}, \tag{10}$$

$$\mathrm{JER} = \sum_{k=1}^{\mathrm{K}} \left(J_{i,k} - \mathrm{J}_{i,k-1} \right) / \mathrm{K} \tag{11}$$

B. Comparison Results of Jitter Estimation based on Spike Detection

To evaluate the performance of the proposed jitter estimation based on spike detection, the experiment was performed on four network traces listed in Table 1.

TABLE I. STATISTICS OF NETWORK TRACES

Trace	End-to-end network delay (ms)	STD of network delay (ms)	Maximum jitter (ms)	Network packet loss (%)
A	49.29	26.02	295	0
B	42.17	57.75	392	0
C	48.79	31.97	374	0

In Table I, average of the network delay, standard deviation (STD) of the network delay (which reflects the jitter characteristics for each condition), and maximum jitter (which is the difference between the maximum and minimum delay in the short trace) are depicted. Because we want to focus on the effect of jitter estimation based on spike detection in this section, four network delay traces with the extreme maximum jitter over 250 ms are chosen from the

Internet links of the testbed infrastructure. And the network traces do not carry any network packet loss.

The performance of the proposed jitter estimation method is compared with three methods. The three methods used have been modified from the contents of reference papers and implemented. Method 1 is based on the adaptive gap-based algorithm [7] incorporated with spike detection [3], while Method 2 is based on an adaptive NLMS playout algorithm with delay spike detection [1]. In Method 3, a timing recovery and loss substitution method [16] is combined with modeling the statistics of the interarrival times with the K-Erlang distribution [5]. Four methods commonly incorporate the packet loss concealment [10].

Table II shows the experimental results on jitter estimation error and late loss rate for each trace. PM denotes the proposed algorithm.

TABLE II. PERFORMANCE OF THE JITTER ESTIMATION

Trace	Method	Jitter estimation error(ms)	Late loss rate (%)
A	PM	34.4	1.06
	Method 1	68.6	2.37
	Method 2	51.8	1.73
	Method 3	60.5	2.02
B	PM	46.1	1.87
	Method 1	86.9	3.22
	Method 2	67.5	2.53
	Method 3	77.3	2.89
C	PM	36.9	1.46
	Method 1	74.6	2.55
	Method 2	55.8	2.11
	Method 3	66.2	2.43

As shown in Table III, the proposed jitter estimation based on spike detection achieves smaller jitter-estimation errors and late loss rate overall than Method 1, Method 2, and Method 3.

C. Comparison Results of Adaptive Signal Reconstruction and Playout Control

The six network delay traces that we collected from the Internet links for the performance evaluation are listed in Table III.

TABLE III. STATISTICS OF NETWORK TRACES

Trace	End-to-End Network Delay (ms)	STD of Network Delay (ms)	Maximum Jitter (ms)	Network Packet Loss (%)
1	25.38	7.46	48	1.93
2	24.82	8.30	48	3.99
3	34.80	13.37	152	3.97
4	47.17	17.88	195	1.99
5	79.98	29.62	363	1.96
6	78.22	31.22	371	3.97

STD, standard deviation

The performance of our proposed method is compared with three methods that have been modified from the contents of reference papers and then implemented. Method 1 is based on an adaptive normalized least mean square playout algorithm with delay spike detection [1] and packet loss concealment [16]. In Method 2, the time-scale modification and loss substitution method [5] are combined with modeling the statistics of the inter-arrival times with the K-Erlang distribution [5].

Table IV depicts the experimental results of the four methods. M1, M2, and PM denote Method 1, Method 2, and the proposed method, respectively.

TABLE IV. EXPERIMENTAL RESULTS OF THE FOUR METHODS

Trace	Method	Speech (Sampling Rate: 16 kHz)			Music (Sampling Rate: 16 kHz)		
		TBE (ms)	JER (ms)	PESQ Score	TBE (ms)	JER (ms)	PESQ Score
1	M1	42.74	18.46	2.857	43.47	17.54	2.825
	M2	35.23	23.57	3.635	36.27	24.29	3.156
	PM	**26.48**	**14.17**	**3.902**	**28.67**	**14.23**	**3.432**
2	M1	44.51	18.62	2.703	45.64	20.09	2.512
	M2	31.66	23.41	3.243	33.36	23.27	2.753
	PM	**25.54**	**14.13**	**3.682**	**27.93**	**13.83**	**3.285**
3	M1	43.28	29.54	2.534	40.78	30.21	2.072
	M2	45.39	29.36	2.877	46.48	29.37	2.532
	PM	**40.36**	**21.47**	**3.432**	**39.74**	**21.38**	**3.125**
4	M1	41.54	29.85	2.821	41.36	28.02	2.356
	M2	50.15	29.19	3.267	49.11	28.11	2.943
	PM	**43.32**	**21.93**	**3.753**	**44.54**	**21.48**	**3.557**
5	M1	50.56	77.82	2.902	46.31	75.31	1.747
	M2	85.55	78.24	3.082	87.26	78.04	2.675
	PM	**59.86**	**68.18**	**3.414**	**66.19**	**68.84**	**3.237**
6	M1	51.81	78.51	2.679	45.11	78.55	1.747
	M2	85.47	78.18	2.605	78.78	78.25	2.248
	PM	**52.21**	**69.51**	**3.126**	**62.36**	**69.09**	**2.846**

TBE, total buffering delay; JER, jitter estimation error; PESQ, perceptual evaluation of speech quality; M1, Method 1; M2, Method 2; PM, proposed method

Table IV shows that our proposed method (PM) outperforms the reference methods M1, and M2 in medium jitter, high jitter, 2 % packet loss rate, and 4 % packet loss rate. The highest PESQ scores were achieved using the PM in traces 1. The PESQ scores of speech samples were higher than those of music samples.

In particular, as the jitter levels or packet loss rates increase, the PESQ scores decrease. The performance difference becomes more significant, showing the clear advantage of the PM. The PM is well-suited for operating with low buffering delay against dynamic changes in network conditions, and handling various loss patterns.

IV. CONCLUSIONS

In this paper, we proposed and evaluated an adaptive signal reconstruction and playout control for enhancing VoIP speech and music quality. The proposed fully receiver-based enhancing algorithm enables users to deliver high-quality voice or music using the combined signal reconstruction and playout control. Experimental results confirm that the proposed method achieves higher PESQ values than the other methods and is suitable for use in any practical mobile VoIP system.

In the future, we will apply the proposed method to advanced teleconferencing applications running on Smart TV.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency), NIPA-2013-H0301-13-4005

REFERENCES

[1] A. Shallwani, “An adaptive playout algorithm with delay spike detection for real-time VoIP,” Electrical and Computer Engineering, IEEE CCECE 2003, Canadian Conference on, Montreal, Quebec, Canada, vol. 2, May, 2003, pp. 997-1000.

[2] B. Sat and B. W. Wah, "Analyzing voice quality in popular VoIP applications," IEEE Multimedia, vol.16, issue 1, January, 2009, pp.46-59.

[3] Y. Liang, N. Farber, and B. Girod, “Adaptive playout scheduling and loss concealment for voice communication over IP networks,” IEEE Transactions on Multimedia, vol. 5, no. 4, April, 2001, pp. 532-543.

[4] V. P. Bhute and U. N. Sharawankar, “Speech packet concealment techniques based on time-scale modification for VoIP,” International Conference on Computer Science and Information Technology, Singapore, Singapore, August, 2008, pp. 825-828.

[5] H. Li, G. Zhang, and W. Kleijn, “Adaptive playout scheduling for VoIP using the K-Erlang distribution,” The 2010 European Signal Processing Conference, Algborg, Denmark, August, 2010, pp. 1494-1498.

[6] H. Sanneck, A. Stenger, K. B. Younes, and B. Girod, “A new technique for audio packet loss concealment,” Global Telecommunications Conference, London, UK, November 1996, pp. 48–52.

[7] H. Sanneck, A. Stenger, K. B. Younes, and B. Girod, “A new technique for audio packet loss concealment,” Global Telecommunications Conference, London, UK, November 1996, pp. 48–52,

[8] J.-H. Chen, “Packet loss concealment for predictive speech coding based on extrapolation of speech waveform,” ACSSC 2007 Conference Record of the Forty-First Asilomar Conference on Signal, Systems and Computers, California, USA, November 2007, pp. 2088-2092.

[9] J. Pinto and K. J. Christensen, “An algorithm for playout of packet voice based on adaptive adjustment of signalspurt silence periods,” 24th Conference on Local Computer Networks, Lowell, USA, vol. 5, October 1999, pp. 224-231.

[10] S. Chi and B. F. Womack, “QoS-based optimal adaptive playout buffer scheduling using the packet arrival distribution,” IEEE MILCOM, California, USA, October 2009, pp. 15-19.

[11] J. Aragao Jr. and G. Barreto, “Novel approaches for online playout delay prediction in VoIP applications using time series models,” Computers & Electrical Engineering, vol. 36, issue 3, May 2010, pp. 536-544

[12] D. Florencio and L.-W. He, “Enhanced adaptive playout scheduling and loss concealment techniques for Voice over IP networks,” 2011 IEEE International Symposium on Circuits and Systems, Rio de Janeiro, Brazil, May 2011, pp. 129-132.

[13] S. L. Ng, S. Hoh, and D. Singh, “Effectiveness of adaptive codec switching VoIP application over heterogeneous networks,” 2nd International Conference on Mobile Technology, Applications and Systems, Guangzhou, China, November 2005, pp. 7-13.

[14] ITU-T Rec. P.862, “Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality qssessment of narrowband telephone networks and speech codecs,” International Telecommunication Union, Geneva, Switzerland, February 2001.

[15] N. Aoki, “A VoIP packet loss concealment technique taking account of pitch variation in pitch waveform replication,” Electronics and Communications in Japan, vol. 89, no. 3, March 2006, pp. 1-9

[16] S. V. Andrsen, W. B. Kleijn, and P. Sorqvist, “Method and arrangement in a communication system,” U.S. Patent 7 321 851, 2008.