IEICE TRANS. COMMUN., VOL.E89–B, NO.6 JUNE 2006

PAPER

Adaptive Voice Smoothing with Optimal E-Model Method for VoIP Services

Shyh-Fang HUANG[†], *Nonmember*, **Pao-Chi CHANG**[††], *and* **Eric Hsiao-kuang WU**[†††a)], *Members*

SUMMARY VoIP, one of emerging technologies, offers high quality of real time voice services over IP-based broadband networks; however, the quality of voice would easily be degraded by IP network impairments such as delay, jitter and packet loss, hereon initiate the presence of new technologies to help solve out the problems. Among those, playout buffer at the receiving end can compensate for the jitter effects by its function of tradeoff between delay and loss. Adaptive smoothing algorithms are capable of the dynamical adjustment of smoothing size by introducing a variable delay based on the use of the network parameters so as to avoid the quality decay problem. This paper introduces an efficient and feasible perceived quality method for buffer optimization to achieve the best voice quality. This work formulates an online loss model which incorporates buffer sizes and applies the ITU-T E-model approach to optimize the delay-loss problem. Distinct from other optimal smoothers, the proposed optimal smoother can be applied for most codecs and carries the lowest complexity. Since the adaptive smoothing scheme introduces variable playback delays, the buffer re-synchronization between the capture and the playback becomes essential. This work also presents a buffer re-synchronization algorithm based on silence skipping to prevent unacceptable increase in the buffer preloading delay and even buffer overflow. Simulation experiments validate that the proposed adaptive smoother achieves significant improvement in the voice quality.
key words: *adaptive voice smoother, VoIP, buffer re-synchronization, delay/loss trade off, E-model*

1. Introduction

The rapid progress of the development of IP-base network has enabled numerous applications that deliver not only traditional data but also multimedia information in real time. Being the future trend of new generation network, the all ALL-IP network is expected to integrate all heterogeneous wired and wireless networks and provide seamless worldwide mobility, where one revolution of the new generation Internet applications will help to realize VoIP services for people to talk freely around through the mobile-phones, the desktops and VoIP telephones at any time and any place. Unfortunately, the IP-based networks do not guarantee the available bandwidth and the constant delay jitters (i.e., the delay variance) for real time applications. In other words, given a varying traffic load and differing routing paths under congestion conditions, individual transmission delays for a given flow of packets in a network may suffer continual change, so that the packet network delays for a continuous series of intervals (i.e. talkspursts) at the receiver may not be the same (i.e. constant) as the sender. In addition, a packet delay may be introduced by the signal hand-out or the difference of bandwidth transportation in wireless/fixed networks.

In terms of delay sensitive applications, a dominating portion of packet losses might be caused by delay constraint. A late packet that arrives after a delay threshold determined by playback time is treated as a lost packet. A tight delay threshold not only degrades the quality of playback but also reduces the effective bandwidth because a large fraction of delivered packets are dropped. In fact, delay and loss are normally not independent of each other. In order to reduce the loss impact, a number of applications utilize the smoothing technique in which buffers are adopted to reduce the voice damage caused by loss packets. However, a large buffer will induce excessive end-to-end delay and deteriorate the multimedia quality in interactive real-time applications. Therefore, a tradeoff is required between increased packet loss and buffer delay to achieve satisfactory results for playout buffer algorithms.

In the past, the works on the degradation of the voice quality considered the effect of packet loss, but not that of packet delay. Among the literature on predicting delays, the use of Pareto distribution in [1] is to compute the distribution parameters and rebuild the new distribution to predict the next packet delay, and the use of neural network models is to learn traffic behaviors [2]. Either the use of Pareto distribution or a neural network model requires relatively high complexity or a long learning period. Therefore, we consider the smoothers [3]–[9] instead with the expect to employ statistical network parameters, i.e. loss, delay and talkspurt, because they are related with voice characteristics and have the significant influence to the voice quality. They also detect delay spike in traffic and are able to quickly calculate the required buffer size to keep the quality as good as possible.

The E-model is a computational model, standardized by ITU-T in G.107, G.109 and G.113 which uses the various transmission parameters to predict the subjective quality of packetized voice. Unfortunately, the E-model is complicated to analyze in the optimization process. R. Cole and J. Rosenbluth [10] first proposed an alternative study to apply a simplified E-model, which is based upon observed transport measurements in the VoIP gateways and the trans-

Manuscript received June 8, 2005.
Manuscript revised September 9, 2005.
[†]The author is with Electrical Engineering, National Central University, Chang-Li City, 320 Taiwan.
[††]The author is with Communication Engineering, National Central University, Chung-Li City, 320 Taiwan.
[†††]The author is with Computer Science & Information Engineering, National Central University, Chung-Li City, 320 Taiwan.
a) E-mail: hsiao@csie.ncu.edu.tw
DOI: 10.1093/ietcom/e89–b.6.1862

port paths. Authors indicated that more pattern cases were required for the simplified E-model method as traces to enhance the validation. Atzori and Lobina [11], and L. Sun and E. Ifeachor [12] proposed to utilize another simplified E-model, it functions in a way by considering loss and delay altogether so as to set a dijitter time, which is also the optimal playout delay derived by a dynamic programming-based solution. However, the usability and the accuracy of a simplified E-model are limited by non-typical traffic patterns.

For perceptual-based buffer optimization schemes for VoIP, voice quality is evaluated as a key metric since it represents user's perceived QoS. However, it requires an efficient, accurate and objective way to optimize perceived voice quality. To consider the well-defined delay and loss impairments of the E-model, we employ a complete E-model for the quality optimization to obtain optimal perceived voice quality.

In a packet switching network, if without a resynchronization scheme, a playback clock with a minor frequency error will eventually cause a buffer overflow or an underflow at the receiving end. The overflow packets are usually discarded due to the finite buffer size and the real-time requirement. This discontinuity caused by discarded packets might create an unpleasant effect to the playback quality because the lost packets could be the important part of signals. This effect is more serious for audio signals than video signals because human ears are more sensitive to the continuity of sounds than human eyes.

The contributions of this paper are three-fold: (i) A new method optimizing voice quality for VoIP is easily applied to codecs which were well-defined in the ITU-T E-model. (ii) Different from the other optimal smoothers, our optimal smoother has the lowest complexity with *O(n)*. (iii) A feasible scheme is introduced to solve the buffer resynchronization problem.

2. Related Work

The performances of the proposed playout approach are compared with the other approaches, in particular, the linear filter, Spike Detection (SD) algorithm [3]–[12] referred by most people for non-dynamic programming-based solutions. A delay spike is defined as a sudden and significant increase of network delay in a short period which is often less than one round-trip. This algorithm adjusts the smoothing size, i.e. playback delay, at the beginning of each talk-spurt. The results of this algorithm are therefore compared to the results obtained herein.

The SD Algorithm in [3] estimates the playout time p_i of the first packet in a talk-spurt from the mean network delay d_i and the variance v_i for packet i as

$$p_i = t_i + d_i + \gamma v_i \tag{1}$$

where t_i represents the time at which packet i is generated at the sending host and *gamma* is a constant factor used to set the playout time to be "far enough" beyond the delay estimate such that only a small fraction of the arriving packets could be lost due to late arrival. The value of $\gamma = 4$ is used in simulations [3]. The estimates are recomputed each time a packet arrives, but only applied when a new talk-spurt is initiated.

The mean network delay d_i and variance v_i are calculated based on a linear recursive filter characterized by factors α and β

$$\begin{cases} If(n_i > d_{i-1}) \Rightarrow \\ \quad \begin{cases} d_i = \beta d_{i-1} + (1-\beta) n_i \\ v_i = \beta v_{i-1} + (1-\beta) \left| d_{i-1} - n_i \right| \end{cases} \\ If(n_i \leq d_{i-1}) \Rightarrow \\ \quad \begin{cases} d_i = \alpha d_{i-1} + (1-\alpha) n_i \\ v_i = \alpha v_{i-1} + (1-\alpha) \left| d_{i-1} - n_i \right| \end{cases} \end{cases} \tag{2}$$

where n_i is the end-to-end delay introduced by the network and typical values of α and β are 0.998002 and 0.75 [3], respectively. The decision to select α and β is based on the current delay condition. The condition $n_i > d_{i-1}$ represents network congestion ($SPIKE\_MODE$) and the weight β is used to emphasize the current network delay. On the other hand, $n_i \leq d_{i-1}$ represents stable traffic in the network, and α is used to emphasize the long-term average.

In estimating the delay and variance, the SD Algorithm uses only two values α and β that are simple but may not be adequate, particularly when the traffic is unstable. For example, an under-estimated problem is when a network becomes spiked, but the delay n_i is just below the d_{i-1}, the SD Algorithm will judge the network to be stable and will not enter the $SIPKE\_MODE$.

3. Adaptive Smoother with Optimal Delay-Loss Trade off

One of the greatest challenges to VoIP is voice quality and one of the keys to acceptable voice quality is the quality of the transmission channel. Therefore, it is essential for the passive monitoring agent to track the performance of transmission channel. A common and standardized method for measuring the voice transmission quality is ETSI Technical Report ETR-250 (E-model), which allows the E-model a well accepted candidate for appropriate voice quality monitoring. The proposed optimal smoother is derived using the E-model to trade off between the delay and loss. This method is accomplished by three steps: first, to build the traffic delay model and the loss model; second, to calculate the delay and loss impairments of the E-model accordingly by the delay and the loss models; third, to maximize the E-model rank *R* and thus the delay and loss optimized solution is obtained.

In this study, voice packets are assumed to be generated at a constant packet rate. Current voice codecs used in standard VOIP (H.323 or SIP) systems, e.g., G.711, G.723.1 and G.729, generally fit this assumption although the packet size may be different when the voice is inactive.

3.1 E-Model Description

In the E-model, a rating factor R represents voice quality and considers relevant transmission parameters for the considered connection. It is defined in [13] as:

$$R = R_o - I_s - I_d - I_e\_eff + A \tag{3}$$

R_o denotes the basic signal-to-noise ratio, which is derived from the sum of different noise sources which contain circuit noise and room noise, send and receive loudness ratings. I_s denotes the sum of all impairments associated with the voice signal, which is derived from the incorrect loudness level, non-optimum sidetone and quantizing distortion. *Id* represents the impairments due to delay of voice signals, that is the sum of Talker Echo delay (*Idte*), Listener Echo delay (*Idle*) and end-to-end delay (*Idd*). *Ie_eff* denotes the equipment impairments, depending on the low bit rate codecs (*Ie, Bpl*) and packet loss (*Ppl*) levels. Finally, the advantage factor A is no relation to all other transmission parameters. The use of factor A in a specific application is left to the designer's decision.

3.2 The Delay and Loss Models in E-Model

For perceived buffer design, it is critical to understand the delay distribution model as it is directly related to buffer loss. The characteristics of packet transmission delay over Internet can be represented by statistical models, which are found to follow Exponential distribution for Internet packets (for an UDP traffic) has been shown to be consistent with an Exponential distribution [14]. In order to derive an online loss model, the packet end-to-end delay is assumed as an exponential distribution with parameter $1/\mu$ at the receiving end for low complexity and easy implementation. The CDF of the delay distribution $F(t)$ can also be represented by [15]

$$F(t) = 1 - e^{-\mu^{-1}t} \tag{4}$$

and the PDF of the delay distribution $f(t)$ is

$$f(t) = \frac{dF(t)}{dt} = \mu^{-1}e^{-\mu^{-1}t} \tag{5}$$

where μ is defined as the inverse of the average mean delay.

In a real-time application, a packet loss that is solely caused by extra delay can be derived from the delay model $f(t)$. Figure 1 plots the delay function $f(t)$, which shows that when the packet delay exceeds the smoothing time; the delayed packet is regarded as a lost packet. The loss function $l(t_b)$ can be derived as

$$l(t_b) = \int_{t_b}^{\infty} f(t)\mathrm{d}t = (-e^{-\mu^{-1}t})\,|_{t_b}^{\infty} = e^{-\mu^{-1}t_b} \tag{6}$$

3.3 Optimization on E-Model

The delay and loss factors over transmission have greater impacts to the voice quality than the environments or equipments. To simplify the optimization complexity, and investigate on delay and loss impairments, we make three assumptions in a communication connection as the following: (i). The circuit noise, room noise and terminate signals will not change. (R_o and I_s are fixed). (ii). An echo delay in the Sender/Receiver will not change. (*Idte* and *Idle* are fixed). (iii). A codec will not change (I_e is fixed). In [13], R is rewritten as Eq. (7)

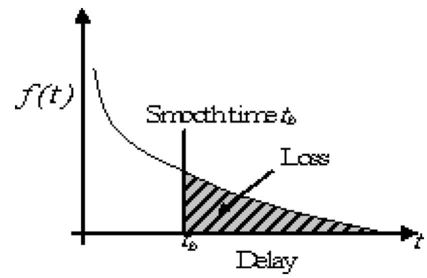


Fig. 1 The relation of smoothing delay and loss.

$$R = (R_o - I_s - Idte - Idle + A) - Idd - I_e\_eff \tag{7}$$

where *Idd* is approximated by

$$Idd = 25\left\{(1+X^6)^{1/6} - 3\left(1+\left[\frac{X}{3}\right]^6\right)^{1/6} + 2\right\},$$
$$X = \frac{ln\left(\frac{t}{100}\right)}{ln(2)},\ when\ T_a > 100\,\mathrm{ms} \tag{8}$$

and $Idd = 0$ when $T_a \leq 100$, and

$$I_e\_eff = I_e + (95 - I_e)\cdot\frac{Ppl}{Ppl + Bpl} \tag{9}$$

Factors I_e and *Bpl* are defined in [16] and T_a is one-way absolute delay for echo-free connections.

Due to the three assumptions above, the optimization process can be concentrated on the parameters of *Idd* and I_e_eff. Equation (7) is derived to yield Eq. (10)

$$R = Constant - 25\left\{(1+X^6)^{1/6} - 3\left(1+\left[\frac{X}{3}\right]^6\right)^{1/6} + 2\right\}$$
$$-(95 - I_e)\cdot\frac{Ppl}{Ppl + Bpl},\ when\ t > 100 \tag{10}$$

The differential equation $\frac{dR}{dt}$ is assigned to zero to maximize R to yield the best quality. According to Eq. (6), the loss probability $Ppl=e^{-u_b}$, so we can get

$$R' = 25 * \left\{(1+X^6)^{-5/6}X^5X' - \left[1+\left[\frac{X}{3}\right]^6\right]^{\frac{-5}{6}}\left(\frac{X}{3}\right)^5 X'\right\}$$
$$-\frac{\lambda e^{-ut}\cdot Bpl}{Bpl^2 + e^{-2ut} + 2\cdot Bpl\cdot e^{-ut}} = 0,$$
$$X = \frac{log\frac{t}{100}}{log2},\ X' = \frac{1}{log2\cdot t} \tag{11}$$

Table 1 The relation of smoothing time and arrival rate.

smoothing time	**μ (1/sec)**	**smoothing time**	**μ (1/sec)**
t_1=110 ms	μ_1=9.71	t_9=190 ms	μ_9=16.95
t_2=120 ms	μ_2=10.64	t_{10}=200 ms	μ_{10}=17.86
t_3=130 ms	μ_3=11.49	t_{11}=210 ms	μ_{11}=18.52
t_4=140 ms	μ_4=12.35	t_{12}=220 ms	μ_{12}=19.61
t_5=150 ms	μ_5=13.33	t_{13}=230 ms	μ_{13}=20.41
t_6=160 ms	μ_6=14.08	t_{14}=240 ms	μ_{14}=21.28
t_7=170 ms	μ_7=14.93	t_{15}=250 ms	μ_{15}=22.22
t_8=180 ms	μ_8=15.87		

The solutions for t are difficult to get directly from Eq. (11) since it contains the complex polynomial and exponential function. Therefore, we will try to solve the best smoothing time t with a numerical approach. This is the major reason why recent solutions for E-model optimization all employing the simplified E-model for optimizations may be difficult to find a realistic optimal solution in the formula. They proposed an optimal algorithm and used dynamic programming tools (e.q. MATLAB) to assist calculating an optimal solution from two dimension variables (i.e. delay and loss). These methods will cost a great amount of computing time to find an optimal solution and will not be suitable and practical for real-time VoIP systems. Different from dynamic programming approach for existing optimal algorithms, we will try to solve the best smoothing time t with a complexity $O(n)$ approach with the help of the observed three condition constraints.

We notice the following three conditions. (i). In Eq. (8), when the smoothing time t ms, *Idd* is zero (no delay impairment). It implies a smoother should set the minimum smoothing delay to 100 ms to prevent the greatest packet loss. (ii). The maximum end-to-end delay of 250 ms [17], [18] is acceptable for most user applications to prevent serious voice quality destruction. (iii). For a common low bit rate codec, like G.723.1 and G.729, the frame rate is 30 ms and 10 ms, respectively, so the gcd(10,30) is 10 ms. Based on the above conditions, we can study the fifteen cases, $t_1 = 110$ ms, $t_2 = 120$ ms,..., $t_{15} = 250$ ms, to calculate the respective correspondence, $\mu_1, \mu_2, \ldots, \mu_{15}$, by the numerical analysis in Eq. (11) with an error less than 0.001. Table 1 shows the smoothing time t corresponding to μ. We can observe that as μ increases, the smoother will enlarge the smoothing time to smooth the late packets. According to Table 1, the proposed smoother will calculate the current $\mu(\mu_{current})$ at the beginning of each talk-spurt and search for a minimum n which satisfies $\mu_n \geq \mu_{current}$. The optimal smoothing time will be $100 + n * 10$ ms to keep the optimal voice quality.

4. Buffer Re-Synchronization

A crucial factor for a smoother to work correctly is the synchronization between the capture and the playback.

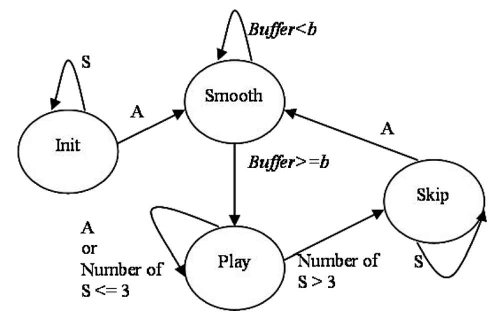


Fig. 2 Buffer re-synchronization machine.

This section proposes a buffer re-synchronization machine (BRM) to help synchronization and the clock drift analysis of re-synchronization to validate the effectiveness.

4.1 Buffer Re-Synchronization Machine

This work proposes a synchronization scheme that segments audio signals by detecting silences. The mismatch between the capture and the playback clocks is solved by skipping silences at the receiving end. The duration of the silent period may be shortened negligibly for degrading the quality of playback. An active packet contains voice-compressed data, whereas a silent packet does not. Skipping some silent packets will not significantly degrade the quality of the voice, but can efficiently prevent the buffer from overflowing. Notably, p (could be adjusted) continuous silent packets could be utilized to separate different talkspurts.

Figure 2 depicts the buffer re-synchronization algorithm. The Init-state, Smooth-state, Play-state and Skip-state are used to represent the voice conference initialing, the buffer smoothing, the buffer playing out, and the silent packets skipping, respectively, and "A" and "S" represent an active packet and a silent packet, respectively.

In the Init-state the buffer waits for the first arriving packets to initialize a voice conference. If Init-state receives an "S," it stays in Init-state; otherwise when an "A" is received, the Smooth-state is activated to smooth the packets. In the Smooth-state, the smoothing time b is computed by applying the optimal adaptive smoother algorithm dynamically. When the buffer smoothing time is over b, the Play-state is activated; otherwise it stays in Smooth-state for smoothing. In the Play-State the packet is fetched from the buffer and played-out. In fetching process, when it encounters three consecutive S packets, implying that the talk-spurt can be ended, the buffer re-synchronization procedure then switches to the Skip-state. In the Skip-state, if "A" is fetched from buffer, it means the new talk-spurt has begun, and then the skips would remain as silent packets in the buffer, and switch to the Smooth-state to smooth the next talk-spurt. Otherwise, if "S" is fetched from buffer, it implies current talk-spurt is not ended and will be decoded to play out at the same state.

With the above four-state machine, the smoother can smooth the packets at the beginning of the talkspurt to avoid buffer underflow in the Smooth-state and skip the silent packets at the end of the talkspurt to prevent the overflow in the Skip-state.

4.2 Effectiveness of Re-Synchronization

To demonstrate the effectiveness of re-synchronization machine for buffer overflow, we analyze the clock inconsistence constraint as the following. C_s and C_r represent the sender clock (frame/sec) and the receiver clock, respectively, and M_a and M_s denote the mean active packets and mean silent packets in a talkspurt, respectively. The buffer overflow caused by the clock inconsistence (difference) will occur when C_s is larger than C_r condition. $C_s - C_r$, the difference value by subtracting the receiver clock from the sender clock, represents the positive clock drift between the sender and the receiver. Therefore, $(C_s - C_r) * ((M_a + M_s) * frame\_time)$ represents the mean extra buffer size caused by the positive clock drift for a mean talkspurt time. In order to distinguish the consecutive talkspurts, p silent packets should be utilized in any cases. Therefore, the smoother can skip on $M_s - p$ silent packets and resynchronizes with the following talkspurt. When the re-synchronization machine satisfies

$$(C_s - C_r) * ((M_a + M_s) * frame\_time) \leq (M_s - p), \quad (12)$$

the buffer overflow caused by the positive clock drift will not occur.

5. Simulation

5.1 Simulation Configuration

A set of simulation experiments are performed to evaluate the effectiveness of the proposed adaptive smoothing scheme. The OPNET simulation tools are adopted to trace the voice traffic transported between two different LANs for a VoIP environment. Ninety personal computers with G.729 traffics are deployed in each LAN. The duration as well as the frequency of the connection time of the personal computers both follow Exponential distributions. Ten five-minute simulations were run to probe the backbone network delay patterns, which were used to trace the adaptive smoothers and compare the effects of the original with the adapted voice quality latter.

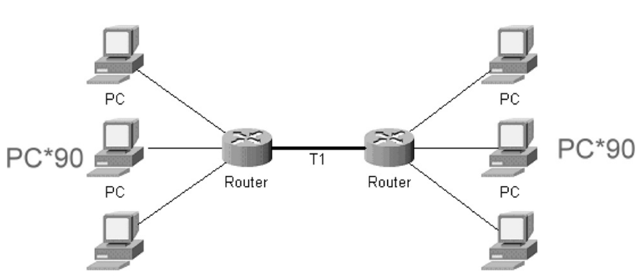
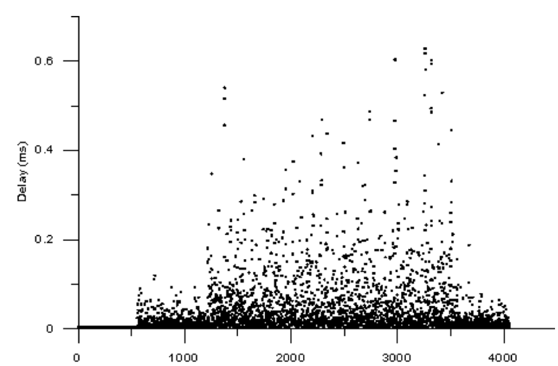


Fig. 3 The simulation environment of VoIP.

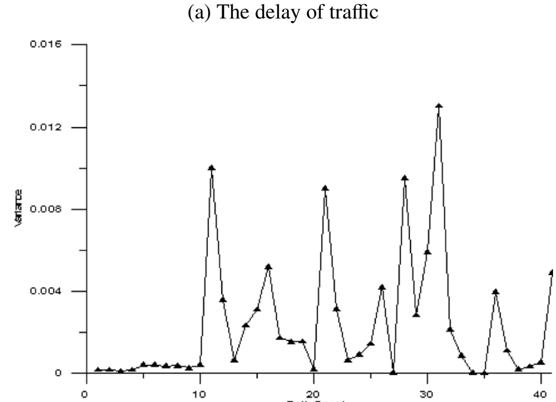
Figure 3 shows the typical network topology in which a T1 (1.544 Mbps) backbone connects two LANs, and 100 Mbps lines are connected within each LAN. The propagation delay of all links is assumed to be a constant value and will be ignored (the derivative value will be zero) in the optimization process. The buffer size of the bottlenecked router is assumed to be infinite since the performance comparison of adaptive smoothers will be affected by overdue packet loss (over the deadline) and not affected by the packet loss in router buffer. The network end-to-end delay of a G.729 packet with data frame size (10 bytes) and RTP/UDP/IP headers (40 bytes) is measured for ten five-minute simulations by employing the OPNET simulation network. Table 2 summarizes the simulation parameters. Figures 4(a) and 4(b) list one of the end-to-end traffic delay patterns and the corresponding delay variances for VoIP

Table 2 Simulation parameters.

Attribute	Value
Numbers of PC in one LAN	90 PCs
Codec	G.729
Backbone	T1(1.544 Mbps)
LAN	100 Mbps
Propagation delay	Constant
Router buffer	Infinite
Packet size	50 bytes



(a) The delay of traffic



(b) The variance of traffic

Fig. 4 VoIP traffic pattern.

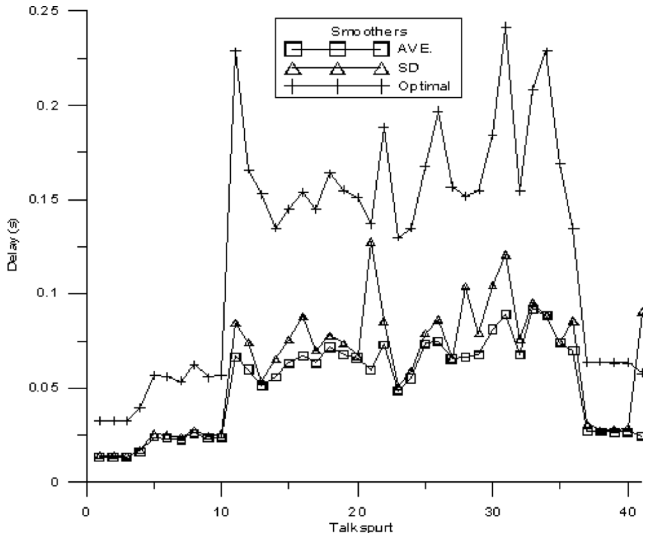


Fig. 5 The predicting time of smoothers.

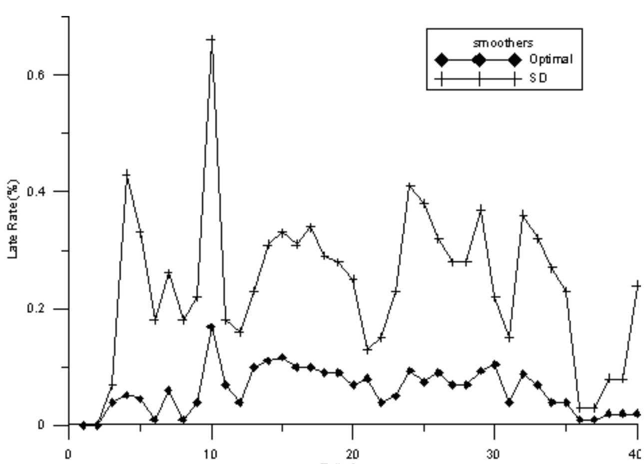


Fig. 6 The packet loss rate of smoothers.

traffic observed at a given receiver.

5.2 Predicted Smoothing Time and Loss Rate in Smoothers

In this section the accuracy of the predicted end-to-end delay time and loss rate among these smoothers are compared. The mean delay is used to observe the traffic pattern in particular. In Fig. 5 and Fig. 6, we can observe that the predicted time of the SD smoother is very close to the mean delay and the loss rate is higher than optimal smoother. The SD smoother uses a large value of fixed β to deal with various traffic conditions and emphasize a long-term mean delay d_{i-1}, so the predicted delay will be close to the mean delay. A better choice for n_i is probably the maximum delay in the last talkspurt that can sufficiently represent the worst case of current network congestion and avoid an under-estimated delay.

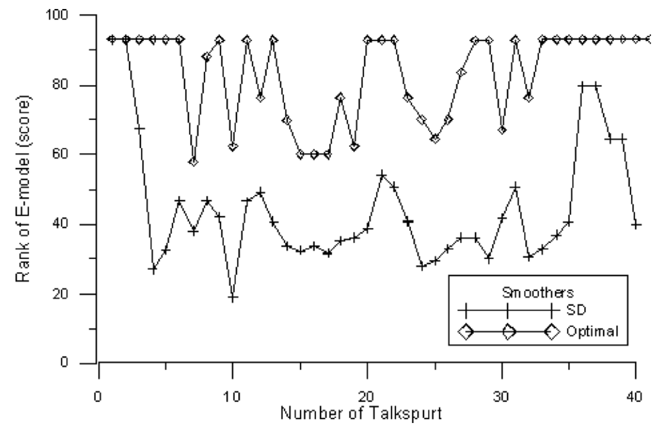


Fig. 7 The quality scores of smoothers.

5.3 Voice Quality in Smoothers

The test sequence is sampled at 8 kHz, 23.44 seconds long, with the experiments of verbal tests in English and Mandarin sentences spoken by male and female. Figure 7 shows the E-model score *R* of the voice quality. It shows that the optimal method has the significant improvement in the voice quality over SD smoother, because our proposed optimal smoother has been proved to optimize with the delay and loss impairments in a transmission planning of the E-model.

References

[1] V. Brazauskas and R. Serfling, "Robust and efficient estimation of the tail index of a one-parameter pareto distribution," North American Actuarial Journal available at http://www.utdallas.edu/˜serfling, April 2000.

[2] P.L. Tien and M.C. Yuang, "Intelligent voice smoother for silence-suppressed voice over internet," IEEE J. Sel. Areas Commun., vol.17, no.1, pp.29–41, Jan. 1999.

[3] R. Ramjee, J. Kurise, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," Proc. IEEE INFOCOM, pp.680–686, June 1994.

[4] D.R. Jeske, W. Matragi, and B. Samadi, "Adaptive play-out algorithms for voice packets," Proc. IEEE Conf. on Commun., vol.3, pp.775–779, 2001

[5] J. Pinto and K.J. Christensen, "An algorithm for playout of packet voice based on adaptive adjustment of talkspurt silence periods," Proc. IEEE Conf. on Local Computer Networks, pp.224–231, Oct. 1999.

[6] Y.J. Liang, N. Farber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications," Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, vol.3, pp.1445–1448, 2001.

[7] A. Kansal and A. Karandikar, "Adaptive delay estimation for low jitter audio over Internet," IEEE GLOBECOM, vol.4, pp.2591–2595, 2001.

[8] A.K. Anandakumar, A. McCree, and E. Paksoy, "An adaptive voice playout method for VOP applications," IEEE GLOBECOM, vol.3, pp.1637–1640, 2001

[9] P. DeLeon and C.J. Sreenan, "An adaptive predictor for media playout buffering," Proc. IEEE Conf. on Acoustics, Speech, and Signal Processing, vol.6, pp.3097–3100, 1999.

[10] R. Cole and J. Rosenbluth, "Voice over IP performance monitoring," Journal on Computer Commun. Review, vol.31, pp.9–24, April

2001.
[11] L. Atzori and M. Lobina, "Speech playout buffering based on a simplified version of the ITU-T E-model," IEEE Signal Process. Lett., vol.11, no.3, pp.382–385, March 2004.
[12] L. Sun and E. Ifeachor, "New models for perceived voice quality prediction and their applications in playout buffer optimization for VoIP networks," Proc. ICC, pp.1478–1483, June 2004.
[13] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning," March 2003.
[14] J.C. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," Journal of High-Speed Networks, vol.2, pp.305–323, Dec. 1993.
[15] K. Fujimoto, S. Ata, and M. Murata, "Statistical analysis of packet delays in the Internet and its application to playout control for streaming applications," IEICE Trans. Commun., vol.E84-B, no.6, pp.1504–1512, June 2001.
[16] ITU-T Recommendation G.113, "Transmission impairments due to speech processing; Appendix I: Provisional planning values for the equipment factor le and packet-loss robustness factor Bpl," May 2002.
[17] C. Perkins and O. Hodson, "Options for repair of streaming media," Network Working Group, RFC2354, June 1998.
[18] H. Zlatokrilov and H. Levy, "Packet dispersion and the quality of voice over IP applications in IP networks," IEEE INFOCOM, pp.1170–1180, 2004.

Shyh-Fang Huang was born in Taiwan on December 30, 1971. He received the B.S. degree in Computer Science from Chung Hua University in 1995 and the M.S. degree in Computer Science & Information Engineering from National Central University in 1997. This is his last year on Ph.D. in National Central University. His current interests are in the area of ATM/IP protocols and video/voice applications.

Pao-Chi Chang was born in Taipei, Taiwan, in 1955. He received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree from Stanford University, California, 1986, all in electrical engineering. From 1986–1993, he was a research staff member of the department of communications at IBM T.J. Watson Research Center, Hawthorne, New York. At Watson, his work centered on high speed switching system, efficient network design algorithms, network management, and multimedia teleconferencing. In 1993, he joined the faculty of National Central University, Taiwan. His current interests are in the area of speech/image coding, watermarking, and video coding over high speed networks and wireless communications.

Eric Hsiao-kuang Wu received his B.S. degree in computer science and information engineering from National Taiwan University in 1989. He received his Master and Ph.D. in computer science from University of California, Los Angeles (UCLA) in 1993 and 1997. He is an Associate Professor of Computer Science and Information Engineering at National Central University, Taiwan. His primary research interests include Wireless Networks, Mobile Computing and Broadband Networks. He is a member of IICM (Institute of Information and Computing Machinery) and IEEE.