社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE

VoIP 端末の QoS 測定

小口 和海 † ‡ Wenyu Jiang ‡ Henning Schulzrinne ‡

† 三菱電機（株）情報技術総合研究所 〒247-8501 鎌倉市大船 5-1-1

‡ Department of Computer Science, Columbia University 1214 Amsterdam Avenue, New York, NY, 10027 U.S.A.

E-mail: † koguchi@isl.melco.co.jp, ‡ {wenyu, hgs}@cs.columbia.edu

あらまし VoIP (Voice over IP) は音声処理を行う端末とパケット転送を行うネットワークを用いて実現されるサービスである。我々は複数の VoIP 端末を用いてエンドツーエンドでの品質を測定した。特に mouth-to-ear (M2E) 遅延、クロックスキューの大きさ、パケット損失時の VoIP 端末の挙動について測定を行った。測定の結果、M2E 遅延は主に受信側端末に依存していること、LAN 環境において H/W による VoIP 端末の場合、M2E 遅延は 45-90ms であり、S/W による VoIP 端末の場合、M2E 遅延は 65ms-400ms であった。H/W による VoIP 端末は 20ms 間隔の 2 連続パケット損失に対して補完を行っていた。

キーワード IP テレフォニー, VoIP, QoS, エンドツーエンド, 遅延, ジッタ

QoS Measurement of VoIP End-points

Kazuumi Koguchi † ‡ Wenyu Jiang ‡ Henning Schulzrinne ‡

† Information Technology R&D Center, Mitsubishi Electric Corporation 5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501

‡ Department of Computer Science, Columbia University 1214 Amsterdam Avenue, New York, NY, 10027 U.S.A.

E-mail: † koguchi@isl.melco.co.jp, ‡ {wenyu, hgs}@cs.columbia.edu

Abstract VoIP (Voice over IP) is a service that requires synergy between the underlying network for transport and the end-points responsible for voice processing. We evaluate the end-to-end quality and performance of several VoIP end-points. In particular, we focus on the following aspects: mouth-to-ear (M2E) delay, clock skew and behavior under packet loss. Our measurement results show that M2E delay depends mostly on the receiving end-point, and when hardware IP phones act as receivers, they achieve low average M2E delay (45-90ms) in a LAN environment. For software VoIP clients as receivers, their average M2E delays range from 65ms to over 400ms. We find that all tested hardware IP phones support some form of packet concealment and it works well for up to two consecutive losses at 20 ms packet intervals.

Keyword IP Telephony, voice over IP, quality of service, end-to-end, delay, jitter

1. Introduction

Recently VoIP (Voice over IP) services has become popular and several kinds of VoIP end-points have been commercialized. There are extensive literature on netrwork quality of service (QoS), such as in the areas of Diffserv [1] and Intserv [2]. However, little has been done on studying the QoS of VoIP end-points. VoIP is a service that requires synergy between the network for transport and the end-points responsible for voice processing. Depending on the implementation, such as what playout algorithms these end-points use, whether they are hardware or software based, their performance may differ dramatically. Therefore it is worth to measure and compare QoS using VoIP end-points.

In this paper, we evaluate the QoS of a number of VoIP end-points, including hardware based IP phones and software based end-points. We examined the following QoS aspects, mouth-to-ear (M2E) delay, clock skew and behavior under packet loss.

The rest of the paper is organized as follows. Section 2 describes our experiment setup. Section 3 presents the measurement results and section 4 concludes the paper and lists future work.

2. Experiment Setup

2.1 Measurement system configuration

Our measurement system configuration is shown in Fig. 1. We measure mouth-to-ear delay by recording both the original audio and the receiver's output audio in a two-channel (stereo) mode simultaneously. Using a fork cable, the original audio is output to both an VoIP end-point and a PC for recording. The end-point is connected to another end-point through LAN. The receiver's output is also connected to the PC for recording. After recording we measure delay between the original audio and receiver's output. We use a software which calculates a most likely delay offset based on auto-correlation in the frequency domain. We have confirmed its precision is within 1ms by using an audio mixer which can insert a known delay offset.

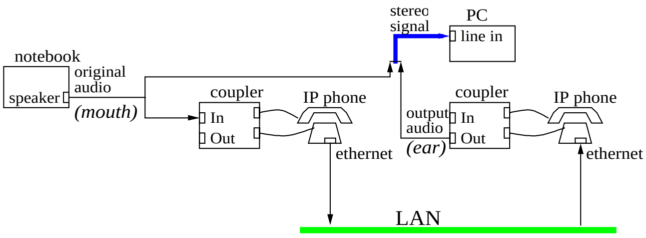

Fig. 1 Measurement System Configuration

2.2 End-point devices

We evaluate several IP phones which are shown in Table 1. Phone (d) is a 1-line PSTN/IP gateway. sipc [3] is a SIP [4] user agent and developed by IRT lab in Columbia university. Phone (e), (f) and (g) are software VoIP clients. We installed (e) and (f) on a AMD K7 PC with Windows 2000/XP dual-boot, and a Pentium-3 notebook with Windows XP. Phone (g) is a software client which can call from PC to PSTN. In addition, we used mobile phone to measure delay to PSTN for reference.

Table 2 shows some parameters in the experiments.

Table 1 List of tested end-points

End-point	type/platform
phone (a)	hardware
phone (b)	hardware
phone (c)	hardware
phone (d)	gateway, hardware
sipc	Solaris, Ultra10
phone (e)	Win 2000/XP(K7)& Win XP(P3)
phone (f)	Win 2000/XP(K7)& Win XP(P3)
phone (g)	Win NT(P2)
GSM phone	GSM 1900 US

Table 2 Parameters in the experiments

End-point	codec	silence suppression	packet interval
phone (a)	G.711 μ-law	N	20ms
phone (b)	G.711 μ-law G.729	Y Y	20ms 20ms
phone (c)	G.711 μ-law	N	20ms
phone (d)	G.711 μ-law	N	30ms
sipc	G.711 μ-law	N	20ms
phone (e)	G.711 μ-law	Y	20ms
phone (f)	G.723.1	Y	30ms
phone (g)	?	Y	60ms

3. Measurement Results

3.1 Mouth-to-ear delay without jitter

Fig.2 shows an example plot of M2E delay over time. The notation "experiment 1-2" means part 2 of call no.1 and "experiment 2-1" means part 1 of call no.2. Between call no.1 and call no.2 we hang up and call up. There is a short pause between part 1 and part 2 to save part 1's audio to disk. We transmitted the same audio data for each experiment.

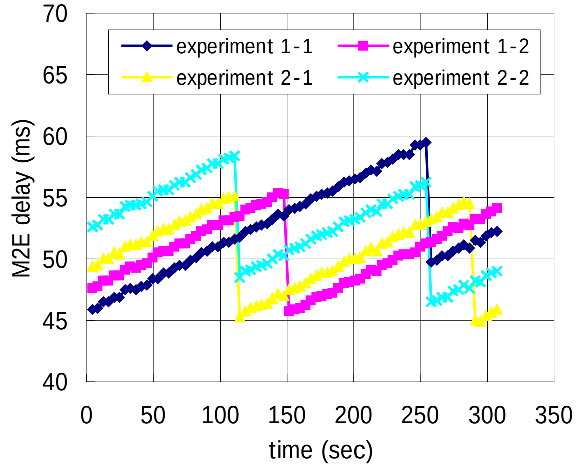

Fig. 2 M2E delay from phone (a) to (b)

In experiment 1-1, the M2E delay from phone (a) to (b) reaches a peak of about 59 ms at the time of 254 seconds, then drops to 49 ms immediately. The highly linear trends is due to clock skew. Similar linear trends and drops in M2E delay are in other experiments.

We investigated the recording files to check how the audio was in Fig.2 case and found there were long silence gaps in the original audio near the drop points. This concurs with the convention that playout delay should only be adjusted at the beginning of a new talk-spurt [5], an event represented by an RTP [6] packet with the marker (M) bit set to 1. Although phone (a) doesn't have capability of silence suppression, hence it will never set its M-bit to 1, phone (b) is still able to adapt the playout delay irrespective of the M-bit and the adjustment occurs during a silence gap.

Fig.3 shows M2E delay in the reverse direction of Fig. 2 case. The delay slope is opposite to Fig.2 case and delay changes occur more often. Fig.3 also shows when phone (b) transmits Mbit=1 packet. We can say delay adjustment occurs when a new talk spurt starts. Because phone (b) does not send packets after detecting silence, phone (a) is possible to suffer from playout buffer underflow and we think it causes to increase delay when a new talk spurt starts.

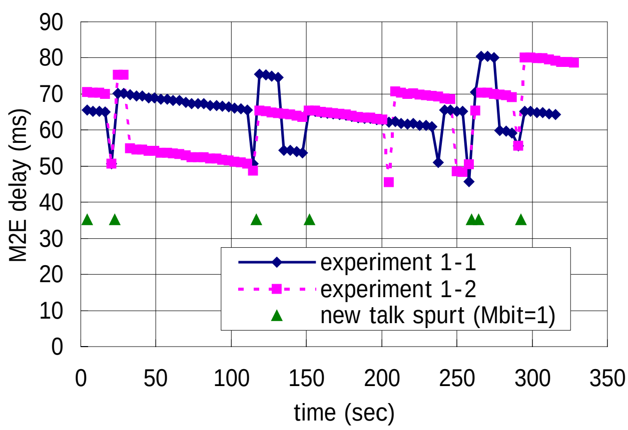

Fig. 3 M2E delay from phone (b) to (a)

Table 3 summarize all test pairs' average M2E delay. The number in the parentheses are the difference (range) between the highest and lowest average M2E delay among all calls for the same pair of end-points. A low range indicates high repeatability, and this is clearly true for Table 3 in most cases. All IP phones achieve a delay below 90ms, and in most cases below 65ms. Phone (c) to (a) has the lowest average delay of 46.6ms. Between two phone (b)s, delay under G.729 (98.7ms) is nearly 20 ms higher than under G.711 (75.8ms), clearly due to additional compression delay for G.729.

Between the IP phones, due to the low delay, it is not very clear whether the sender or receiver plays a more dominant role in M2E delay. However, the role becomes evident when sipc is the receiver. Its M2E delay is consistently above 200ms irrespective of the sender.

Table 3 Average M2E delays (H/W phones and sipc)

		Receiver				
		(a)	(b)	(c)	(d)	sipc
Sender	(a)	—	51.4ms (2.5ms)	55.1ms (2.9ms)	47.1ms (3.1ms)	223ms (8.7ms)
	(b) G.711	63.0ms (3.8ms)	75.8ms	74.0ms (8.6ms)	78.1ms (8.4ms)	206ms (12.3ms)
	(c)	46.6ms (4.1ms)	63.0ms (8ms)	—	57.6ms (1.8ms)	230ms (63.4ms)
	(d)	85.8ms (16.3ms)	77.6ms (4.0ms)	72.5ms (10.7ms)	—	—
	sipc	52.4ms (3.1ms)	59.8ms (0.9ms)	64.2ms (16.4ms)	—	—
	(b) G.729	—	98.7ms	—	—	—

The dominant role of the receiver is more evident in Table 4. For example, phone (e) (Win XP, P3, notebook) to phone (e) (Win XP, K7, pc) achieves an average delay of 120ms. However, when the receiver switches to phone (e) (Win 2K, K7, same pc), the delay consistently drops to 68.5ms, indicated by the small range in Table 4. This confirms that the receiver dominates the M2E delay.

The M2E delay of phone (g) is near or above 300ms. We measured the round-trip time (RTT) between the PC and the PSTN gateway. The RTT are usually 5-30 ms. Therefore the 300 ms M2E delay is not caused by the network, but by the client or the gateway.

Table 4 Average M2E delays (PC clients and GSM)

end-point A	end-point B	A → B	B → A
phone (e) (Win XP, K7)	phone (e) (Win XP, P3)	109ms (0.8ms)	120ms (44.6ms)
phone (e) (Win 2K, K7)	phone (e) (Win XP, P3)	96.8ms (5ms)	68.5ms (10.1ms)
phone (f) (Win 2K, K7)	phone (f) (Win XP, P3)	401ms (46.9ms)	421ms (6.8ms)
phone (g) (Win NT, P2)	PSTN	288ms (77.3ms)	371ms (12.2ms)
mobile phone (GSM)	PSTN	115ms (4.8ms)	109ms (5.1ms)

According to our experiments, the delay between mobile phone and PSTN is around 110ms. Therefore, hardware based IP phones (a)-(d) and PC client phone (e) achieve lower or equivalent delay in a LAN environment.

3.2 Clock skew

Clock skew can cause playout buffer underflow or overflow after a certain period of time, resulting in occasional choppy audio. In our experiments, all the end-points are able to adjust the playout delay whenever clock skew causes the M2E delay to go too high or too low, such as in Fig. 2, 3 and irrespective whether the sender supports silence suppression. However, for phone (a) and (c), they appear to suffer from playout buffer underflow even when there is no packet loss or delay jitter. Because such events usually occur when the delay is being adjusted, we think they are caused by the playout algorithm.

We analyze the rate of clock skew. Table 5 shows average clock drift rates. Symmetric drift rates are observed in A->B and B->A direction.

Table 5 Average clock drift rates (in ppm)

		Receiver				
		(a)	(b)	(c)	(d)	sipc
Sender	(a)	—	55.4	41.2	43.3	-333
	(b) G.711	-55.2	-0.4	-12.1	-11.8	-381
	(c)	-40.9	12.7	—	2.8	-380
	(d)	-43.1	11.7	-0.8	—	—
	sipc	343	403	376	—	—
	(b) G.729	—	-0.4	—	—	—

The drift rate depends on the crystal oscillator used to drive the voice circuitry, but its magnitude usually falls within 100 ppm, with 25 ppm being typical [7]. Between IP phones, the magnitude of the combined drift is within 60 ppm. However for sipc it is always higher than 300 ppm, far beyond 100ppm. Between two phone (b)s, there is almost no clock skew, presumably because they use the same type of oscillators.

3.3 Behavior under packet loss

When a packet is lost in the network, if no retrans-mission or forward error correction [8] is used, the receiver is responsible for repairing the lost packet with some approximation, a procedure known as packet loss concealment (PLC) [9]. The simplest method is silence substitution, by replacing the lost waveform with silence. It however produces the worst quality. A better option is packet repetition, by repeating waveform in the last packet. A further improvement is extrapolation of last packet. The best method is interpolation, if the packets before and after the loss are received in time. All these methods usually fade the voice gradually to prevent any annoying repetitive or buzzing sound.

We evaluated packet loss behaviors on the hardware based IP phones (phone (a), (b) and (c)). Packet loss is simulated by a UDP relay program that connects from sipc to an IP phone. For easy verifiability, the relay program generates deterministic bursty packet losses. For example, it can generate 5 consecutive losses for every 100 packet received (denoted as 5/100). We tested 1/100, 3/100, 5/100 and 1/20 for all three phones.

We examine the output waveform and notice that every phone can compensate two (three for phone (b)) consecutive losses. Phone (c) repeats the last waveform during a packet loss and it does some smoothing. Phone (a) and (b) uses at least extrapolation or interpolation. When the loss pattern is 1/100, there is no audible distortion for any of these IP phones.

Table 6 summarizes the behaviors of the IP phones. It also shows the relative quality of PLC among the phones, that is, how audible is the audio distortion. The quality rating is given by the authors by listening to the output audio, but the difference between the three levels (almost inaudible, audible and very audible) are very clear.

Table 6 PLC behaviors

IP phone	PLC	loss tolerance	loss pattern	
			3/100	1/20
(a)	extrapolation only?	2 Packets	audible	very audible
(b)	extrapolation or interpolation	3 packets	almost inaudible	almost inaudible
(c)	packet repetition with edge smoothing	2 packets	audible	very audible

Fig. 4 shows how the M2E delay of the phone (c) changes over time under different loss patterns.

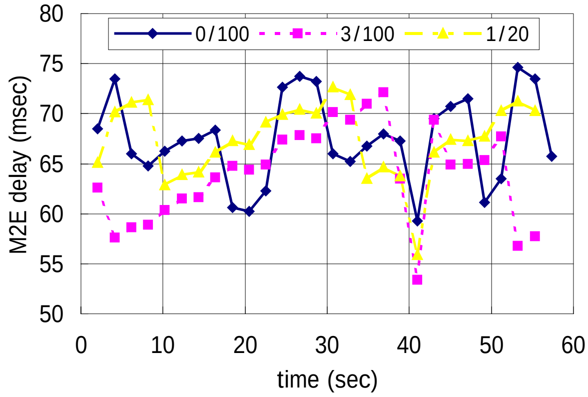

Fig. 4 M2E delay under packet loss, sipc to phone (c)

We find that: first, all of the tested phones implement a PLC better than silence substitution. Second, their PLC usually works for up to two or three consecutive losses at 20ms interval, then the voice immediately fades to silence. This is acceptable because repairing more than three consecutive losses becomes extremely difficult and it may cause side effect like buzzing sound. Thirdly, PLC does not increase the M2E delay in any affirmative way.

3.4 Mouth-to-ear delay under network jitter

We performed an initial test on how jitter affects the M2E delay. We used the same UDP relay program used for packet loss tests, and inserted delay based on a packet trace. We used one of the typical packet traces collected between a university machine and a PC with cable modem. It is well known that cable modem uplink exhibits high jitter. The 99% delay is 43.8 ms for the downlink trace we used, and 93.6 ms for the uplink trace, nearly 50 ms higher. Fig. 5 shows the experiment results with the phone (b) as the receiver. The average delay is 38.4 ms higher in the uplink case. This increase is slightly less than the 50 ms difference in the 99 % delay between the two traces. In addition, we do not notice any distortion in the output audio. Therefore, phone (b) does a good job of playout buffering and results in almost no late loss.

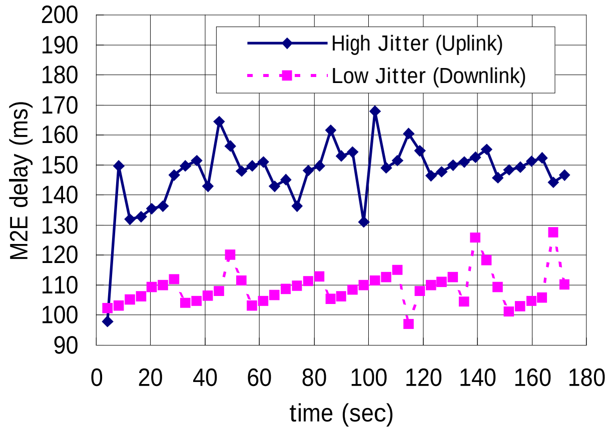

Fig.5 M2E delay under network jitter, sipc to phone (b)

4. Conclusions and Future Work

We have performed a number of measurements evaluating various VoIP endpoints. We find that the mouth-to-ear (M2E) delay depends more on the receiver. Most hardware based IP phones can achieve low M2E delay, typically below 65 ms in average. For software VoIP clients, their average M2E delays range from 65ms to over 400ms. All tested end-points compensate for clock skew, but some appear to have playout buffer underflow occasionally.

We find that all tested IP phones support some form of packet concealment and it works well for up to two consecutive losses at 20 ms packet intervals. Our preliminary test about jitter shows that an IP phone is able to tolerate typical cable modem uplink jitter.

There are several items to investigate further. For example, measuring other VoIP end–points; analyzing behavior when packet reordering occurs; analyzing jitter behavior on more VoIP end-points.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, An architecture for differentiated service, RFC 2475, IETF, Dec. 1998.

[2] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification, RFC 2205, IETF, Sept. 1997.

[3] IRT Lab, Columbia University, sipc home page, http://www.cs.columbia.edu/IRT/software/sipc/.

[4] J.Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, SIP: Session Initiation Protocol, RFC 3261, IETF, June 2002.

[5] R. Ramjee, J. Kurose. D. Towsley and H. Schultzrinne, Adaptive playout mechanisms for packetized audio applications in wide-area networks, IEEE Infocom, pp.680-688, Tronto, Canada, June 1994.

[6] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, RTP: a transport protocol for real-time applications, RFC 1889, IETF, Jan. 1996.

[7] Semiconductor, D. DS1553: 64k NV Y2KC Timekeeping RAM Manual, Tech. rep., Dallas Semiconductor, 1999.

[8] J. Rosenberg and H. Schulzrinne, An RTP payload format for generic forward error correction, RFC 2733, IETF, Dec. 1999.

[9] C. Perkins, O. Hodson and V. Hardman, A survey of packet loss recovery techniques for streaming audio, IEEE Network 12, 5, pp.40-48, Sept. 1998.