

AUTOMATED LEARNING OF PLAYOUT SCHEDULING ALGORITHMS FOR IMPROVING PERCEPTUAL CONVERSATIONAL QUALITY IN MULTI-PARTY VOIP

Zixia Huang, Batu Sat, and Benjamin W. Wah

Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{zhuang21, batusat, wah}@uiuc.edu

ABSTRACT

In this paper, we propose four methods for equalizing the silence periods experienced by users in a multi-party VoIP conversation in order to improve their perceived conversational quality. To mitigate the unbalanced silence periods caused by delay disparities in Internet connections, the playout scheduler at the receiver of each client equalizes the silence periods experienced. Our limited subjective tests show that we can improve the perceptual quality when the network connections are lossy and have large delay disparities. Because it is impossible to conduct subjective tests under all possible conditions, we have developed a classifier that learns to select the best equalization algorithm using learning examples derived from subjective tests under limited network and conversational conditions. Our experimental results show that our classifier can consistently pick the best algorithm with the highest subjective conversational quality under unseen conditions, and that our system has better perceptual quality when compared to that of Skype (Version 3.6.0.244).

Index Terms— Delay equalization, multi-party VoIP, mutual silence, Skype, subjective conversational quality.

1. INTRODUCTION

In a two-party conversation, each client takes turns in speaking and listening, and both perceive a silence duration (called *mutual silence* or MS) in between turns. In a face-to-face setting, both clients have a common perspective of the conversation and experience the same MS as the other. However, when the conversation is carried out over a network with delays, the MSs are perceived as alternating short and long silence durations between turns. This asymmetry is caused by the fact that after A speaks, the silence period experienced by A is governed by the time for A's speech to travel to B (called the *mouth-to-ear delay* or $MED_{A,B}$), the time for B to construct a response (called the *human response delay* of B or HRD_B), and the time for B's response to travel to A ($MED_{B,A}$). In

contrast, after A receives the response from B, the silence period experienced before A speaks is only governed by his/her HRD_A . This asymmetry leads to a degraded perception of interactivity and loss of *conversational efficiency* (CE) [1].

The extension of a VoIP system from two-party to multi-party is not straightforward. The perceived effects of delays in multi-party VoIP is more complex because there may be large disparities in network conditions between any two clients [2]. Hence, each client may experience different MSs and a different perspective from the other clients. In particular, the design of transmission schemes for supporting multi-party VoIP is different from that of two-party VoIP. Unicast transmissions of speech packets from one speaker to all participants at the same time may cause congestion near the speaker. In contrast, a *centralized scheme*, which Skype employs, utilizes a single VoIP client as the host to relay all traffic from the speaker to all participants. This approach may cause both computation as well as network bottlenecks at the relay.

In our previous work [2], we have developed a dynamic *overlay network* (ON) whose parents are fully connected and whose children are each connected to a single parent. This approach balances the trade-offs between end-to-end network delays (which affect conversational dynamics) and packet transmission rates (which affect network congestion). To improve the quality of received speech segments, we have also designed an end-to-end *loss-concealment* (LC) scheme by piggybacking redundant copies of previously transmitted packets in the current packet, and a cooperative *playout scheduling* (POS) scheme that dynamically adjusts the jitter-buffer delay at the receiver. As the paths from different clients to a node can have different characteristics, we use unique jitter buffers for each source client at this node.

Multi-party conversational model. After hearing a speech segment from A in a multi-party conversation, the next speaker B waits for a short HRD_B before responding. As is discussed before, another participant, say C, perceives this switch from A to B differently. Let $MED_{i,j}$ include the sum of link delays between i and j and the delay at the jitter buffer

IEEE INT'L CONFERENCE ON MULTIMEDIA AND EXPO, 2008.

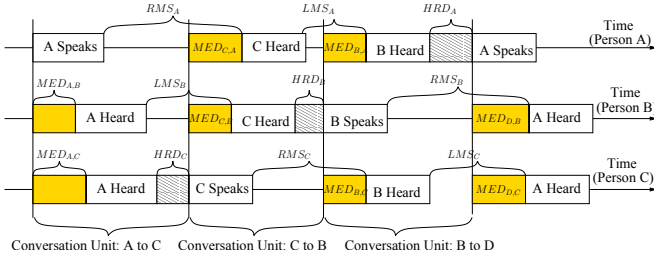


Fig. 1. A multi-party VoIP conversation.

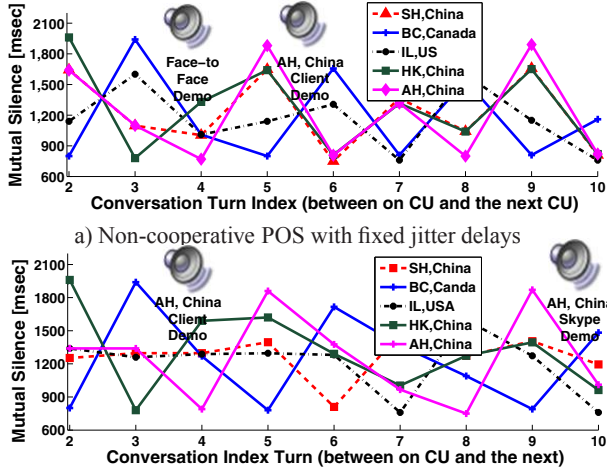


Fig. 2. Disparities in MSs for a 5-node multi-party VoIP conference. (Click the icon for demo.) The average HRD is around 750 ms.

of receiver j . Further, let $MS_C^{A \rightarrow B}$ be the MS experienced by C on the switch from A to B (Figure 1), where:

$$\begin{aligned} MS_A^{A \rightarrow B} &= MED_{A,B} + HRD_B + MED_{B,A} \\ MS_B^{A \rightarrow B} &= HRD_B \end{aligned} \quad (1)$$

$$MS_{k \notin \{A,B\}}^{A \rightarrow B} = MED_{A,B} + HRD_B - MED_{A,k} + MED_{B,k}.$$

Note that the MS perceived by A differs from that of C, since A waits for a response to his/her own speech (similar to two-party), but C is currently just listening. To capture this special case, we define the *respondent MS* (RMS) $MS_i^{i \rightarrow j}$ and the *listener MS* (LMS) $MS_{k \notin \{i,j\}}^{i \rightarrow j}$, where i is the previous speaker and k is neither the previous nor the current speaker.

Variations in MS. Some of the variations in MSs in multi-party VoIP are inevitable. The current speaker experiences HRD (usually the shortest MS) when switching from the last speaker, and RMS (usually the longest MS) when switching to the next speaker. These correspond to the MSs in the two-party case and cannot be reduced without compromising the perceptual quality. This pair of speakers in a particular turn are called the *bottleneck pair*. For example, in Turn 3 in Figure 2, the bottleneck pair consists of the speaker at BC, Canada, and the speaker at HK, China, where the speaker at HK experiences an MS of 770 ms (due to HRD), and the speaker at BC experiences an RMS of 1940 ms (due to 770 ms HRD and round-trip MEDs of 1170 ms). Note that the bottleneck pair changes from one turn to another.

In contrast, the remaining listeners perceive LMS that do not contribute to the bottleneck. Each passive listener belongs to a *non-bottleneck pair* with respect to the speaker in a given turn. Although their MSs may have large variations, they can be equalized by increasing the corresponding delay at each client. The equalized MSs can contribute to improved perceptual conversational quality. As an example, the LMSs of the three passive listeners in Turn 3 have been equalized in Figure 2b. Note that RMS is usually much larger than LMS.

Previously, we have defined *conversational symmetry* (CS) [2] as a metric for capturing the variations in MS in a multi-party VoIP conversation. Here, we slightly modify the definition to capture the variations in LMS and RMS:

$$CS_k = \frac{\max_j MS_k^{i \rightarrow j}}{\min_{j, j \neq k} MS_k^{i \rightarrow j}} \text{ over a past window.} \quad (2)$$

Intuitively, the numerator represents the maximum of the k^{th} curve in Figure 2b, whereas the denominator is the minimum while discounting the minimum term of HRD. Note that CS_k for client k should be 1 in a face-to-face conversation.

Approach. Based on the network conditions observed in Section 2, we present in Section 3 four POS algorithms and study their performance by comparative subjective evaluations. As there are infinitely many network and conversational conditions and conducting offline subjective evaluations to determine the best algorithm under each condition is infeasible, we study the design of a classifier in Section 4, using limited number of offline comparative subjective evaluations to learn user preferences in conversational quality. The classifier learned allows us to choose the best algorithm under unseen network and conversational conditions at run time.

2. NETWORK & CONVERSATIONAL CONDITIONS

Table 1 summarizes the delay, jitter, and loss statistics of seven UDP trace sets collected in PlanetLab (grouped into five classes). The behavior is non-stationary and dynamic over time. To avoid overwhelming the network with excessive traffic, we have collected each group of one-to-five traffic traces at different times and have put them together into a single five-by-five trace set. This approach is valid because the behavior of the five one-to-five trace sets is independent.

We set the one-way *end-to-end delay* of a packet as:

$$T = (t_2 - \Delta t_2) - (t_1 - \Delta t_1), \quad (3)$$

where t_1 and t_2 are the sending and arrival times of the packets according to the local clocks. We synchronize the clock of node i using its local official NTP server and obtain the offset Δt_i . Our approach is valid because the inaccuracy of the local clocks is much smaller than the end-to-end delay of a packet.

Table 2 summarizes the average, minimum, and maximum of the lengths of speech segments and the conversation order of two multi-party social conversations extracted from a television series. One conversation consists of fifteen turns from three females and two males, and the other has thirteen turns from two females and three males.

Table 1. Internet traces collected in 2007 and 2008 (DL: delay; JT: jitter; JT60: jitters larger than 60 ms with respect to mean delay; and LR: loss rate). Delays are classified into low (less than 100 ms), high (larger than 100 ms), and mixed (a combination of both). Similarly, jitters are classified into low (less than 5% in JT60), high (greater than 5% in JT60), and mixed; and losses into low (less than 5%), high (greater than 5%) and mixed. Shaded boxes indicate the parent nodes of the overlay network in each trace set [2].

#	Location	DL/JT/LR (L/H/M)	Avg DL(ms)		JT60(%)		LR(%)	
			Min	Max	Min	Max	Min	Max
1	CA,US	Class 1 L/L/L	45	92	0.2	3.6	0.0	0.1
	IL,US		45	63	0.0	2.4	0.0	0.0
	Germany		28	92	0.0	2.4	0.0	0.2
	MD,US		58	90	2.4	2.6	0.0	0.0
	UK		29	88	0.0	2.5	0.0	0.2
2	NY,US	Class 1 L/L/L	26	52	0.0	0.0	0.0	0.0
	OR,US		25	60	0.0	0.0	0.0	0.0
	TX,US		26	31	0.0	0.0	0.0	0.0
	CA,US		11	39	0.0	0.0	0.0	0.0
	MO,US		17	54	0.0	0.0	0.0	0.0
3	BJ,CN	Class 2 M/L/L	50	284	0.4	0.6	0.0	0.0
	IL,US		120	219	0.0	0.2	0.0	0.0
	Hungary		120	290	0.4	0.7	0.0	0.0
	SH,CN		83	301	0.1	2.8	0.0	0.1
	Taiwan		131	319	0.0	7.5	0.2	0.3
4	SD,CN	Class 2 M/L/L	22	242	0.0	0.9	0.1	1.4
	Japan		70	226	0.0	0.0	0.0	0.5
	TJ,CN		27	244	0.0	0.0	0.0	1.1
	TX,CN		124	165	0.0	0.0	0.0	0.0
	Uruguay		121	242	0.0	0.1	0.0	0.0
5	CA,USA	Class 3 L/L/M	42	178	0.0	0.1	0.0	0.0
	Canada		53	148	0.0	0.0	0.0	3.6
	HK		101	131	0.0	1.3	14.3	17.1
	NH,US		49	129	0.0	0.1	0.0	0.2
	AH,CN		97	194	0.0	0.0	0.0	0.1
6	Canada	Class 4 L/M/L	58	202	0.0	2.2	0.0	0.7
	Inda		248	352	12.2	12.9	3.7	4.2
	CA,US		32	185	0.0	0.8	0.0	0.4
	SC,CN		46	301	0.0	0.0	0.0	0.5
	AH,CN		33	296	0.0	0.0	0.0	0.5
7	BJ,CN	Class 5 L/M/M	104	199	0.1	5.3	1.9	8.6
	UK		88	132	0.0	0.1	0.0	0.4
	TX,US		88	163	0.0	2.9	0.0	2.6
	Canada		64	199	0.0	1.4	0.0	1.1
	SX,CN		107	190	0.0	2.8	0.0	0.0

3. MULTI-PARTY PLAYOUT SCHEDULING

Algorithm 1: Fixed POS. Each client estimates the mean delays from others during call establishment. To accommodate jitters, MED is set to be 40 ms larger than the mean delay.

Algorithm 2: Non-cooperative adaptive POS. Similar to a two-party system that updates its MED at the beginning of each talk-spurt [1], Algorithm 2 selects an MED based on the recently collected delay statistics for each speaker-listener pair. At each decision point, statistics corresponding to a past 10-second window is used to determine the MED that would have allowed 99% of the packets to be in time for playout.

Algorithm 3: Cooperative adaptive POS. We have previously developed a multi-party POS algorithm that considers the difference between the *bottleneck* and the *non-bottleneck* nodes [2], where the former is the node that would experience the longest MS when a particular client is speaking. In most cases, the bottleneck node is the previous speaker, due to the

Table 2. Characteristics of speech segments in two five-party social conversations used in our experiments.

Set	Length (ms)			Conversation Order
	Avg	Min	Max	
1	2222	600	4400	ACABCEDBCDBCDBC
2	1603	630	3350	BACBDECDDBC

inherent structure of RMS in (1).

Algorithm 3 uses the MED found by Algorithm 2 when the listener is the bottleneck. For non-bottleneck listeners, it relaxes the MED to a level larger than that chosen by Algorithm 2. This relaxation is limited to a level where the MS observed by the non-bottleneck client equals to the MS observed by the bottleneck client. The relaxation is calculated based on a heuristic parameter that linearly combines the maximum and the minimum allowed MEDs [2]. It improves the concealment of jitters as well as conversational symmetry, without significant effects on conversational efficiency (CE) [2].

Algorithm 4: Distributed equalization. More aggressive equalizations of LMSs for non-bottleneck nodes will lead to CS closer to 1. However, this may lead to reduced CE because it unnecessarily delays all non-bottleneck nodes, who may become the speaker in the next turn and cannot speak until he/she finishes the current turn. Hence, there is a desirable CS and LMS that will result in the best quality. User feedbacks have shown that the maximum LMS should be less than 1500 ms and that a suitable CS is between 1.3 and 1.7.

Algorithm 4 dynamically adjusts the MS of each listener based on the history of MSs. To accommodate fluctuations in MSs, we estimate a range of MSs $[EMS_{min}, EMS_{max}]$ that cover most of the MSs in a conversation. The algorithm considers three cases. a) If the MS of a listener in the last turn is the same as the RMS and is very large, then its current LMS is usually small as compared to RMS and is set to EMS_{max} . This allows less abrupt changes in MSs from the last turn. b) If the MS in the last turn and the current LMS without adjustment are both less than EMS_{min} , then we set it to EMS_{min} . c) If the previous MS is within the range, then we use the moving average of the previous several MSs that are also within the range. Since our results indicate that the size of the moving window has limited influence on perceptual quality, we set the window size to cover three turns. Note that our method does not depend on the HRD in each turn.

Algorithm 4 can run in a non-cooperative or cooperative fashion. In a non-cooperative approach, one client may set its MS to be unnecessarily large because it applies the algorithm without considering the MSs in other clients. In a cooperative approach, each client broadcasts its history of MSs to other clients at the end of a turn. Based on the listener's estimated MS and by assuming that this client is the next speaker, the strategy predicts the MSs of all listeners in the next turn using (1). We first set $MED_{i,j}$ to be the average end-to-end delay from i to j plus 60-ms jitter delay at the receiver. If the equalized MS in the current turn causes any MS in the next turn to be larger than EMS_{max} , we reduce the current MS to a reasonable level according to the current delay statistics.

4. EVALUATION OF CONVERSATIONAL QUALITY

We use comparative subjective evaluations as well as objective measures because there is no single objective measure that can capture user preferences of conversational quality in multi-party VoIP. We have developed a multi-party VoIP simulator [2] that generates conversations using different POS algorithms under each given set of network traces and pre-recorded conversations encoded by the ITU G.722.2 wide-band codec. It uses a fixed 750ms HRD and employs an adaptive overlay network and link-based loss concealments [2].

The *objective measures* captured include the PESQ (ITU P.862) of each speech segment; the statistics of MSs of the conversation; the MS ratio (MSR), which is the average ratio between the maximum and the minimum of two consecutive MSs; and the conversational efficiency (CE) [2].

The *subjective evaluations* are conducted by subjective tests. Subjects were asked to listen in a random order to pairs of conversations generated by two algorithms using our simulator and were asked to indicate their preferences. We use a simplified version of the comparative category rating (CCR) in ITU P.800 and ask subjects to choose among $\{A \text{ better than } B, A \text{ about the same as } B, A \text{ worse than } B\}$.

To determine the *dominant* opinion between two algorithms under a given condition (with $> 50\%$ probability and a certain level of statistical significance), we model the subjective opinions by a multi-nomial distribution with 3 possible outcomes, assuming the independence of samples. We then conduct hypothesis testing by selectively combining two options and have an equivalent binomial distribution that represents the *for* and *against* probabilities of the opinion. Option i is *dominant* if the following hypothesis is accepted:

$$H_0 : \left(p_i, \sum_{j \neq i} p_j \right) \text{ is drawn from } \textit{binomial}(N, p \geq 0.5) \quad (4)$$

where N is the number of samples. For instance, for 90% (resp., 80% and 70%) significance, 27 (resp. 25 and 24) out of 45 samples need to agree on an opinion.

Since there are infinitely many network and conversational conditions, it is impossible to conduct offline subjective evaluations to cover all cases. For this reason, we design a classifier, similar to that in [3], that uses a limited number of comparative subjective evaluations to learn the user preferences and to generalize them to unseen but similar conditions.

We choose a comprehensive set of traces that span the space of possible conditions in conducting subjective tests. We train an SVM-based classifier [LIBSVM], using the objective measures obtained for two conversations as input and their subjective preference as output. We train 3 SVM classifiers, each learning the mapping between the objective measures and the probability of one of the three options ($\{A > B\}, \{A \approx B\}, \{A < B\}$). Both the subjective and the predicted opinion distributions are processed to determine if one of the opinions is dominant with a prescribed level of statistical significance, based on the number of samples collected.

5. EXPERIMENTAL RESULTS

In generating learning patterns for the classifier, we conducted pair-wise comparisons between two algorithms under the first conversational order in Table 2 and each of the five trace sets 1, 3, 5, 6 and 7. Also included in the learning patterns are the subjective-test results between the multi-party version (3.6) of Skype and Algorithm 4 using trace set 4.

In testing the classifier learned, we used all the trace sets and the same conversational order, as well as the second conversational order in Table 2 and a different part of the trace file in trace sets 5 and 7 (denoted by 5N and 7N).

Table 3. Partial orders of the algorithms and the multi-party Skype (SK) in terms of results on subjective tests and tests of the learned classifier on conversational quality with at least 70% significance.

Trace Set + Conv.	Partial Order (Subjective)				Partial Order (Classifier Output)				
	Algorithms 1-4				Algorithms 1-4 & Skype				
	A1	A2	A3	A4	A1	A2	A3	A4	SK
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1
3	2	2	2	1	3	3	2	1	3
4	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	2
5N	1	1	1	1	1	1	1	1	2
6	2	2	1	1	3	2	1	1	3
7	1	1	1	1	1	1	1	1	2
7N	1	1	1	1	1	1	1	1	2

Table 3 summarizes the partial orders found with at least 70% statistical significance. For trace sets with low delay disparities, losses, and jitters (1, 2 and 4), all five alternatives are statistically equal. For 5, 5N, 7, and 7N, the four algorithms are mutually equal, and each is preferred over Skype. Figure 3 further depicts the results of the learned classifier for trace sets 3 and 6.

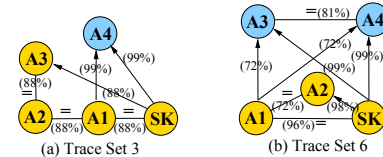


Fig. 3. Partial orders found: an arrow indicates a dominating opinion with the corresponding statistical significance; a missing arrow indicates that a statistically significant relation is not established.

6. REFERENCES

- [1] B. Sat and B. W. Wah, "Playout scheduling and loss-concealments in VoIP for optimizing conversational voice communication quality," in *Proc. ACM Multimedia*, Augsburg, Germany, Sept. 2007, pp. 137–146.
- [2] B. Sat, Z. X. Huang, and B. W. Wah, "The design of a multi-party VoIP conferencing system over the Internet," in *Proc. IEEE Int'l Symposium on Multimedia*, Taichung, Taiwan, Dec. 2007, pp. 3–10.
- [3] B. Sat and B. W. Wah, "Evaluating the conversational voice quality of the Skype, Google-Talk, Windows Live, and Yahoo Messenger VoIP systems," *IEEE Multimedia*, (accepted with minor revision) Dec. 2007.