Exploiting Just-Noticeable Difference of Delays for Improving Quality of Experience in Video Conferencing

Jingxi Xu and Benjamin W. Wah
Dept. of Computer Science and Engineering
The Chinese University of Hong Kong
Statin, Hong Kong
jxxu@cse.cuhk.edu.hk,
bwah@cuhk.edu.hk

ABSTRACT

This paper proposes a novel approach for improving the quality of experience (QoE) of real-time video conferencing systems. In these systems, QoE is affected by signal quality as well as interactivity, both depending on the packet loss rate, delay jitters, and mouth-to-ear delay (MED) that measures the sender-receiver delay on audio signals (and will be the same as that of video signals when video and audio is synchronized). We notice in the current Internet that increasing MED as well as reducing packet rate can help reduce the delay-aware loss rate in congested connections. Between the two methods, the former plays a more important role and applies well to a variety of network conditions for improving audiovisual signal quality, although overly increasing the MED will degrade interactivity. Based on a psychophysical concept called just-noticeable difference (JND), we find the extent to which MED can be increased, without humans perceiving the difference from the original conversation. The approach can be applied to improve existing video conferencing systems. Starting from the operating point of an existing system, we increase its MED to within JND in order to have more room for smoothing network delay spikes as well as recovering lost packets, without incurring noticeable degradation in interactivity. We demonstrate the idea on Skype and Windows Live Messenger by designing a traffic interceptor to extend their buffering time and to perform packet scheduling/recovery. Our experimental results show significant improvements in QoE, with much better signal quality while maintaining similar interactivity.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications—*Computer conferencing, teleconferencing, and video conferencing*; H.1.2 [**Information Systems**]: Models and Principles—*Human factors*

General Terms

Experimentation, Human Factors, Performance

Keywords

Video conferencing, perceptual quality, just noticeable difference

MMSys '13, February 26-March 1, 2013, Oslo, Norway.

1. INTRODUCTION

Video conferencing systems are popular nowadays for social as well as business communications. Free systems like Skype and Windows Live Messenger have attracted many users, but their quality may not be consistent under different network conditions. Commercial systems, in contrast, have more consistent quality but have high initial investments and some have high operating costs.

The quality of a video conferencing system is often measured by its (subjective) *quality of experience* (QoE), which is a function of multiple objective metrics that can reflect the quality perceived by users. Examples of these metrics include the network loss rate, traffic jitter, link delay, conversational condition, and loss concealment mechanisms in the codec. The difficulty in assessing the quality of a video conferencing system is that QoE is highly subjective, and the QoE perceived by a user is a complex and unknown function of the objective metrics [32]. Offline subjective tests can be conducted, but they are expensive to run, and their results may not generalize well to run-time conditions.

To achieve good QoE in a video conferencing system under given network and conversational conditions, it is important to operate the system at an *operating point* with a set of properly chosen parameters. Although finding the best point is difficult, it is possible for subjects to compare in a relative sense the perceptual quality of two operating points and to identify whether one is better than or indistinguishable from another. Such a comparison entails trade-offs among the objective metrics, where some metrics may lead to perceptual improvement while others may cause degradations.

In comparing operating points of existing systems, we have found that many of them are suboptimal. Some overly emphasize interactivity without sufficient attention to signal quality. In some cases, the *mouth-to-ear delay* or MED (the delay from the time one person speaks to the time the speech is heard by the other) is not sufficient to cover the network delay as well as the buffering time to smooth delay jitters and to recover lost packets. Without proper trade-offs between signal quality and interactivity, the overall QoE will be low. Our experimental results have shown that signal quality can be significantly improved if MED is slightly extended. To address this trade-off, it is important to study the extent of extending MED so that the degraded interactivity will not be perceived.

We address this question by using JND (*just noticeable difference*), a concept in psychophysics. JND describes a boundary beyond which the difference in audiovisual quality between the original and the new operating points will be statistically perceptible by humans. In the context of interactive video conferencing under the same network condition, JND defines a range of MEDs from the original MED (operating point) within which humans cannot perceive any difference in interactivity (in a statistical sense) between

the original and the new operating points. For those MEDs in the JND, we are interested in the maximum MED. By increasing the original MED to this maximum MED, we can improve the quality of the system through additional loss concealment mechanisms, without incurring perceptible changes in interactivity.

Problem statement. In this paper, we study the JND in video conferencing under various network and conversational conditions and use it to guide the improvement in QoE of the default operating point of existing video-conferencing systems. This study can help improve the QoE of the default operating points, while assuming a black-box model of the system being studied.

Our approach is to use JND to help improve QoE in video conferencing under various network and conversational conditions. To allow our approach to be generalizable, we study the current Internet conditions from a large set of network traces and investigate when the increase in MED can help mitigate network losses and jitters. There are three contributions of this paper:

- A comprehensive study of network conditions in the Internet with respect to buffering delays and packet sending rates.
- The properties of JND in video conferencing under various network and conversational conditions.
- The improvement of existing proprietary video conferencing systems, without knowing their operating parameters.

The rest of the paper is organized as follows. We first illustrate the suboptimality of the operating points of existing systems in Section 2. Related work is then reviewed in Section 3, followed by an evaluation of network conditions in the Internet in Section 4. In Section 5 we study the properties of JND, with detailed results from subjective tests. We then demonstrate in Section 6 our approach on Skype and MSN and conclude the paper in Section 7.

2. PROBLEM ILLUSTRATION

To understand the trade-offs made in operating points as well as the related issues in existing systems, we first introduce the quality metrics used in this paper.

For evaluating one-way video quality, we adopt a standardized metric called Video Quality Metric (VQM) [30]. By pooling various factors in a linear fashion into an overall metric, VQM demonstrates a higher correlation to the subjective mean opinion score (MOS) than traditional signal quality metrics like PSNR. VQM generally maps MOS into $[0.0, 1.0]$ range, with a smaller value representing better subjective quality and $VQM = 0$ implying a lossless quality. On the other hand, for evaluating one-way audio quality, we adopt an International Telecommunication Union (ITU) standard called Perceptual Evaluation of Speech Quality (PESQ) [4] that uses a human-speech model to capture the various factors affecting perceptual quality. Again, it has a high correlation to perceptual MOS. The range of PESQ is $[-0.5, 4.5]$, and the larger the better. Note that both VQM and PESQ can only be used in offline tests because they require the original sequence as a reference, in addition to their high computational complexities.

In contrast to audiovisual streaming systems, interactivity is an important quality metric in video conferencing. Interactivity depends on delay, which will change the way the two parties relate to each other in a conversation. When delay in a session is long, each party would find more time waiting for the other to react and their perceptual experience would be degraded. Three metrics can measure interactivity: MED, conversational symmetry (CS) and conversational efficiency (CE) [31].

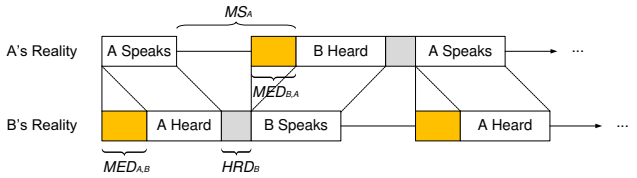


Figure 1: In video conferencing over the Internet, the delays introduced by network propagation and jitter buffers cause users at both sides to have different reality of the conversation. They need more time in waiting for the other party to respond [31].

MED measures the delay before a speech segment is heard by the other side. It consists of the coding time, queuing time, network propagation latency, and playout delay in jitter buffers.

$$MED = t_{\text{encode}} + t_{\text{queue}} + t_{\text{propagation}} + t_{\text{decode}} + t_{\text{playout}}.$$

MED has a direct impact on the interactivity of a conversation. Figure 1 illustrates how MED changes the conversational scenario. Assuming that audio and video data is synchronized, MED changes the conversational structure as follows.

First, $MED_{A,B}$ will delay **A**'s speech in **B**'s reality. After listening to **A**'s speech, **B** will need a short human response delay HRD_B before replying to **A**. Because of $MED_{B,A}$, **B**'s speech will also be heard later in **A**'s reality. As a result, **A** will perceive that **B** is thinking unnaturally long, with a silence period of MS_A:

$$MS_A = MED_{A,B} + HRD_B + MED_{B,A}.$$

Similarly, with $MED_{B,A}$ and $MED_{A,B}$, **A** will discover the asymmetry in the response times; that is, **B**'s thinking time MS_B is longer than **A**'s thinking time HRD_A in **A**'s reality. We measure this asymmetry by CS, which is the ratio the longest MS over the shortest MS in a user's reality:

$$CS = \frac{\max\ MS}{\min\ MS}.$$

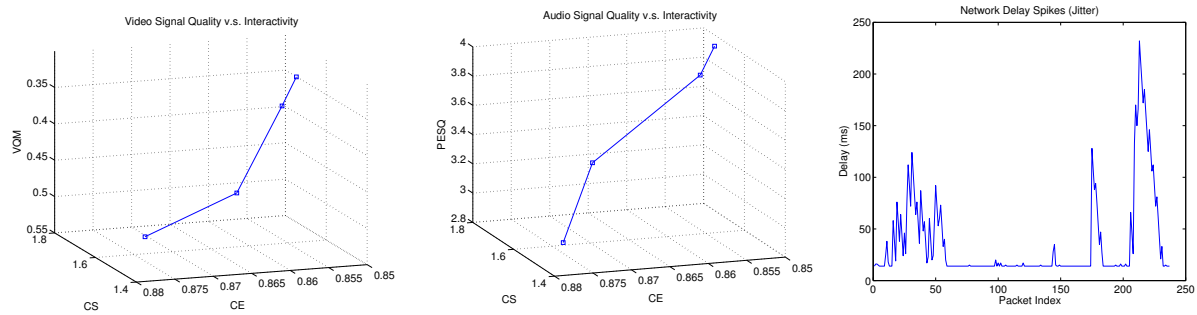
In addition, both parties will find the conversation taking more time than a face-to-face conversation, whose effect can be measured by CE, a metric measuring the ratio of the time in a face-to-face conversation over the time in a video conferencing conversation:

$$CE = \frac{\text{speaking time} + \text{listening time} + \text{thinking time}}{\text{total time for a conversation}}.$$

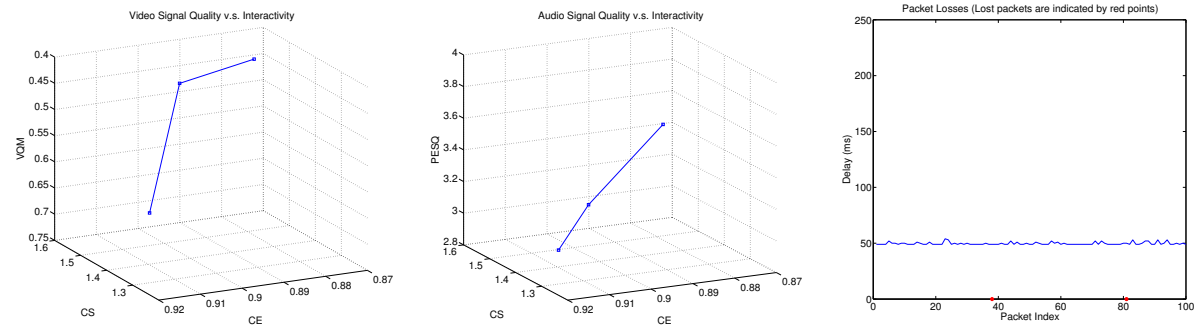
In a face-to-face conversation, (MED, CS, CE) is (0 ms, $1.0, 1.0$). Small MED and CS as well as large CE indicate good interactivity. When MED is increased, CS becomes larger and CE, smaller, which reflect a degradation in interactivity.

In short, we use five metrics to measure QoE: VQM, PESQ, MED, CS, and CE. We do not consider QoS (quality-of-service) metrics like loss rate, delay, jitter and throughput because their effects have been considered in the above metrics. Moreover, some of them may not be available in proprietary systems like Skype (and thus cannot be used to measure quality). For this reason, we do not use QoS metrics like E-model [21] and G.1070 [35] in this paper.

Given the five metrics presented above, trade-offs must be made among them in order to arrive at the best QoE. For example, a larger jitter buffer (which is a part of the overall MED) can mitigate the late arrivals of packets due to network jitters and provide more time for receiving redundant data in recovering lost packets. However, it will cause a longer waiting time for users to receive replies in a conversation. On the other hand, a shorter jitter buffer reduces the chance that a late packet can be received and a lost packet be recovered, and thus degrades the signal quality [31].



a) Skype (under a jittery network condition shown in the third subfigure)



b) MSN (under a lossy network condition shown in the third subfigure)

Figure 2: Video and audio signal qualities versus interactivity in Skype and MSN. The default operating points of these systems are at the bottom left (with MED of 255 ms for Skype and 176 ms for MSN). By increasing MED to the top-right operating point (with MED of 310 ms for Skype and 257 ms for MSN), VQM and PESQ are both improved while CS and CE are both degraded.

Figure 2 shows the relationship between VQM (*respectively* PESQ) versus CS and CE in Skype [33] and Windows Live Messenger (MSN for short) [28] under error-prone connections. The performance was measured by a testbed *RealTalk* (details in Section 6.3) we have developed for improving the performance of proprietary systems. By increasing MED, we can use the additional buffering delay to better protect the multimedia data, leading to better one-way video and audio qualities. At the mean time, as mentioned above, the interactivity of the conversation may suffer.

The results in Figure 2 show that MED affects not only VQM and PESQ but CS and CE as well. Further, the default operating points (MEDs) of both Skype and MSN under these network conditions are suboptimal, leading to poor video and audio qualities. Our results further show that their quality can be improved by increasing MED. An interesting question is, therefore, to what extent can MED be increased, without incurring significant degradation on interactivity that can be perceived by users? We will answer this question by using the concept of JND.

3. RELATED WORK

QoE in video conferencing reflects the subjective quality perceived by humans. Measurements on audio quality [2,15,31], video quality [8, 26, 38], and interaction factors [9] under different conversational conditions are thus of great interest. Due to the complex tradeoffs among the various factors affecting QoE, there is no single metric that can capture the combined effect of one-way perceptual quality and interactivity. Metrics like VQM and PESQ only measure the one-way perceptual quality with a high correlation to MOS, without consideration on interactivity. Guidelines like ITU G.114 [19] focus on interactivity and suggest that MEDs less than 150 ms is desirable and more than 400 ms is unacceptable. One exception is the ITU G.107 E-Model [21] that considers both audio quality and delay by calculating an absolute score for the overall quality. However, it oversimplifies the effects of signal quality and interactivity, since humans like to have short delays when quality is high but can tolerate longer delays when quality is poor. Moreover, the delay requirement depends on the speed and turn frequency of a conversation, which the E-Model does not capture [31]. Likewise, ITU G.1070 [14] addresses the joint effects by considering a linear combination of the effects due to video, audio and delay. For the same reason as above, this cannot precisely measure QoE in video conferencing. An added difficulty in using the E-Model and G.1070 to improve existing proprietary video conferencing systems is that the operating parameters in these systems are not readily available. A general survey on measuring QoE in audio [22] and video [35] can be found in the literature.

Early works on evaluating Skype focus on revealing the internal protocols of Skype. Baset and Schulzrinne [3] have studied Skyps's traffic and discovered its NAT, peer-to-peer organization, session establishment and data encryption schemes. In addition, Suh *et al.* have studied the relayed traffic in Skype [25]. Reverse engineering has been used to decrypt Skype's traffic and find its control mechanisms [5]. It was found that changing Skype's behavior is difficult due to its anti-debugging function. Studies on characterizing Skype's encrypted and overlay traffic as well as methods to filter Skype's traffic have gained a lot of attention [6, 7, 34] due to the overwhelming traffic caused by Skype in the Internet.

To avoid overloading traffic, Skype has introduced congestion control schemes. Many efforts have been made for measuring Skype's performance in congested networks. Cicco *et al.* have performed detailed analysis of Skype's congestion control schemes for both audio [11] and video [12]. Zhang *et al.* have studied

how Skype adjusts its behavior under different stationary network conditions and have shown Skype's TCP friendliness by considering the probability that a user hangs up a call using mean opinion scores (MOS) derived from ITU G.1070 [41]. Besides investigating traffic behaviors, these studies also measured QoS metrics like loss rate, delay, jitter, and throughput, which can partially reflect Skype's performance but cannot lead to an assessment of its QoE.

New schemes have also been proposed for enhancing existing systems. Huang *et al.* have discovered that Skype did not provide appropriate protections for audio packets and have proposed new schemes to improve MOS [16, 17]. They then implemented a new system with audio codecs that Skype has announced to use.

Previous studies emphasize the protocol and the rate-control aspects for improving the signal quality of existing systems but have less focus on delay and interactivity. They have found that short delays are good, without considering the tradeoffs between interactivity and signal quality as well as the extent to which delays can be tolerated. There does not exist any general method that can directly be applied for improving existing proprietary systems like Skype, without knowing their internal designs.

To develop a scheme that is transparent to the internal design of existing video conferencing systems, we propose to examine the effects of delay and extend MED by JND. It is known that JND can be used in many areas for measuring human-related metrics. The famous Weber-Fechner law suggests that JND is proportional to the intensity of stimulation:

$$\frac{JND}{I} = k,$$

where I is the original intensity and k is a constant. On visual QoE, JND has been used to scale video quality in order to increase the efficiency of measurements, both on visual impairment [37] and on compression impairment [23, 36]. A recent work has utilized the viewing-environment-dependent JND in mobile video encoding to improve the signal quality [39]. One work on interactivity in head-mounted display has measured the JND of rendering delays [1], although such delays are different from the two-way delays in interactive video conferencing.

There is no previous work on using JND to adjust delays in video conferencing. JND has been used to minimize the number of subjective tests when finding the optimal operating points in voice-only conferencing [32]. Other works utilize a concept of flicking in adjusting the parameters of video and audio codecs at run time, but do not address the effects of delay in their adjustments [18].

4. NETWORK ENVIRONMENT

The quality of video conferencing highly depends on the network connection. Jitters (delay spikes) and packet losses are two major causes on quality degradations. In this section we study the network environment in the current Internet. To show the importance of using a longer MED, we analyze the cases where a longer MED can conceal network impairments and improve quality.

4.1 Traffic Measurements in the PlanetLab

To have enough traces for analysis, we deploy a UDP probe in the PlanetLab [10], a worldwide overlay network over the Internet for research purpose. With this program, we have collected traces between network nodes, both near and afar. We randomly selected a pair of nodes from a set of 180 nodes and collected a set of traces from 46 different links. To simulate different traffic scales in video conferencing, we collected 1-minute network traces with sending bit rates ranging from 100 kbps to 1000 kbps, a range of bandwidth that a video conferencing system generally uses (see Table 1). We then repeated such 8-minute rounds for two 6-hour periods in September 2012 for the 46 links stated earlier. In total, more than 10,000 1-minute traces have been collected.

Table 1: Parameters used in an 8-minute round of tests

Packet Period (ms)	Packet Size (bytes)	Bit Rate (kbps)	Test Duration (min)
40	500	100	1
20	500	200	1
13	500	308	1
10	500	400	1
8	500	500	1
6	500	667	1
5	500	800	1
4	500	1000	1

Table 2: Classification of raw network conditions

Class	Fraction (%)	Mean Loss Rate (%)	Mean Delay (ms)	Delay Std (ms)	Congested
Good	45	0.02	23	6	N
	22	0.10	94	7	N
	9	0.05	153	2	N
Lossy	7	2.18	45	13	N
Jittery	4	0.09	57	83	Y
	5	0.21	1691	545	Y
	1	0.43	2490	714	Y
Lossy and Jittery	4	11.34	701	257	Y
	2	9.39	63	47	Y
	1	3.22	85	61	Y

To investigate the network behavior from this large set of traces, we use the K-Means algorithm to cluster them. We have tried clustering them into 5, 10, 15 and 20 clusters. Using 10 clusters gives the best division according to the network loss rate, delay, and jitter, while 5 clusters tend to merge them into large sets, and 15 and 20 clusters over-divide them into many clusters with very small differences. Table 2 shows the results when we manually merge the clusters into 4 classes of network conditions according to their loss rate, delay, jitter and whether there is significant congestion (with large increases in loss rate or delay) when bit rate is increased.

A first observation of the results is that the traces in the first class show *good* behavior. In this class, the network is stable without significant congestion, with very few losses and almost no jitters observed. In contrast, the other three classes have either lossy or jittery links or both, which may be triggered by traffic from competing flows, including the flow sent by us. Moreover, in the class with *lossy* behavior, the loss rate is stable no matter what the sending rate is; however, the other two classes have significant increases in either the loss rate or jitters as the sending rate increases.

In all the four classes, it is noticed that the loss rate (*resp.* average delay and jitter) tends to increase or remains unchanged when the sending rate is increased; that is, it is *monotonically non-decreasing* with increasing sending rate. With the traces collected, we validate this assumption by a hypothesis test. In every 8-minute round, we count the times the loss rate (*resp.* average delay and standard deviation of delays) decreases as the sending rate is increased. For example, if the loss rate is decreased only when the sending rate is increased from 100 kbps to 200 kbps and from 400 kbps to 500 kbps, then the count it 2 (out of 7). As small variations in loss rate have negligible effects in video conferencing and exceedingly large average delays are not practical, we discard those cases before the test. This is done by considering loss rates in $[-0.01, 0.01]$ as unchanged, average delays larger than 500 ms as infinity, and jitters larger than 50 ms as infinity. We then perform the *left-tailed* Student t-test on the following null hypothesis using the counts of all the rounds in a 6 hour period (or 45 samples from each link):

- Null hypothesis H_0: The average count on non-increasing loss rates (*resp.* average delays and standard deviations of delay) in a round of test equals 1.

- Alternative hypothesis H_1: The average count on non-increasing loss rates (*resp.* average delays and standard deviations of delay) in a round of tests is less than 1.

With a significance level $\alpha = 0.05$, 94% (*resp.* 91% and 97%) of the links tested reject the null hypothesis. This clearly supports the assumption that the loss rate (*resp.* average delay and jitter) is monotonically *non-decreasing* with increasing sending rate. It is worth mentioning that those cases where the null hypothesis is not rejected have sudden increases in loss rate and/or jitters even when the sending rate is not high. This is most likely caused by congestion introduced by other competing traffic in the network.

In short, although the Internet behaves well most of the time, there are still a sizeable fraction of the traces that suffer from losses and jitters. For those traces, we have found with a high probability that the average loss rate and jitters are monotonically non-decreasing with increasing traffic rate. As there are still a large proportion of links where reducing the traffic rate does not affect the the average loss rate and jitters, traffic rate control is effective only under certain conditions and is not a general strategy for improving connection quality.

4.2 Concealing Impairments by Buffering

The raw network behavior described above is not what a video conferencing system would experience because such a system has real-time constraints in practice and does not wait for late arrivals indefinitely. Consequently, large jitters (and delay spikes) will cause late arrivals in the receiving client and underflows in its play-out buffer. These, together with network losses, will degrade the quality of the signals received. As retransmissions are infeasible in real-time video conferencing, a popular method is forward error correction (FEC) that generates redundant packets for a block of original packets. FEC will incur additional delays because a lost packet can only be recovered after all the necessary original and redundant packets have been received. A general strategy for improving connection quality is, therefore, to increase MED (and thus packet buffering time).

We define UPR in the receiver as the *ratio of unavailable packets within the MED*. These unavailable packets consist of packets arriving later than the prescribed MED, as well packets that cannot be recovered after performing FEC using those packets received in time. Because video and audio codecs both have their maximum tolerable ratios of unavailable packets in transmission, UPR exceeding this threshold cannot be concealed by error-resilient schemes built in the codecs. For this reason, a video conferencing system should always operate within the UPR threshold.

We define UPR as a function of bit rate and MED as follows:

$$UPR = f(loss(bitrate), delay(bitrate), jitter(bitrate), MED)$$
$$= h(bitrate, MED),$$

where f is a function of the link under consideration and is monotonically non-decreasing with loss rate, average delay and average jitters, and monotonically non-increasing with MED; h is a monotonically non-decreasing function of sending rate (according to Section 4.1) and a monotonically non-increasing function of MED (according to the discussion in this section). This behavior is illustrated in Figure 3, which shows the statistics in an 8-minute round for a link from Ohio to North Carolina. Here, MED is assumed to be the buffering time plus the network delay (with negligible encoding, decoding and packetization times). According to monotonicity, the tolerable UPR region is contiguous and located in the lower right-hand corner of the figure. This region represents cases with longer MEDs and lower bit rates.

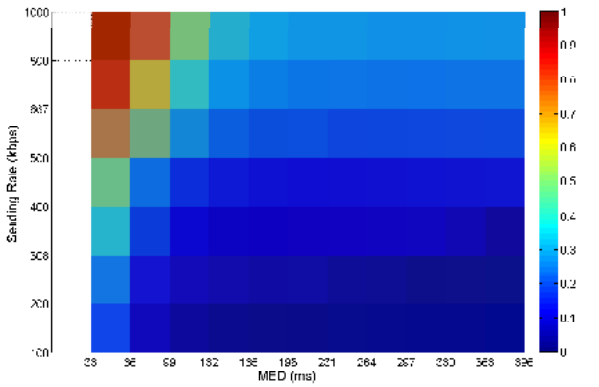


Figure 3: Illustration of UPR at different MEDs and sending rates in an 8-minute round for a connection from Ohio to North Carolina.

Figure 3 further illustrates the actual ratio of packets received ($= 1 - UPR$) at different MEDs and sending rates. Since this ratio directly affects the received signal quality, the figure clearly demonstrates the benefits of using additional buffers (MED) in improving the received signal quality.

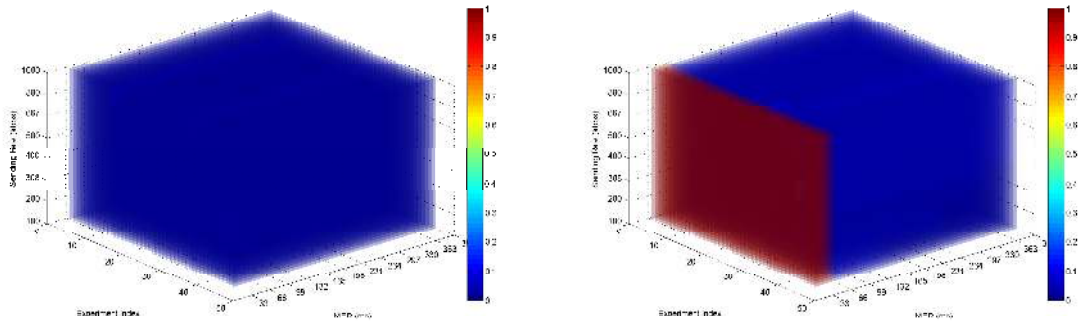
Figures 4 illustrates the UPR over all 8-minute rounds in a 6-hour period for the four classes of network behaviors identified in Table 2. Figure 4(a) illustrates, under a *good* network condition, the very small UPR, which means that no extra buffering and FEC are necessary. The blue blocks indicate the small UPRs and stable network behavior over time. Figure 4(b) illustrates, under a *lossy* network condition, that a slightly increased bit rate will not incur increases in the loss rate and delay (which can be observed from the color distribution over the sending rate). Therefore, FEC is useful for recovering losses in this case. Because receiving the redundant packets in FEC require a longer buffering time, UPR is small only when MED is large enough. Figure 4(c) illustrates, under a *jittery* condition, that increasing MED can significantly reduce UPR. Note that the range of available sending rates is small, as a larger bit rate will incur serious congestion (the red region on the top of the cube). Finally, Figure 4(d) illustrates, under a *lossy and jittery* condition, that increasing MED and reducing sending rate can help reduce UPR, as UPR is reduced significantly from the upper-left corner to the bottom-right corner.

In summary, the properties observed in these PlanetLab traces can help adjust the operating points in video conferencing systems.

- Increasing MED is the most basic method to reduce UPR and provides significant improvements even with a small increase. It is a mechanism that will not change the network behavior and is applicable under any network condition.

- Rate control can help reduce UPR under certain conditions (which is supported by our hypothesis test). In non-congested links (including a large fraction of the traces in Table 2), the average loss rate and delay jitter do not change much when the sending rate is reduced (although the monotonically non-decreasing property still holds). In these cases, rate control does not help improve congestion. Moreover, it is of limited use when the required data rate of codecs is high and the network is congested by other competing flows.

Based on these observations, we propose to increase MED as our main method for reducing UPR in existing video conferencing systems. Rate control has its limitations and may not be as



(a) Good connection: Oklahoma → South Florida

(b) Lossy connection: Massachusetts → France

(c) Jittery connection: California → Minnesota

(d) Lossy and Jittery connection: Ohio → North Carolina

Figure 4: UPR over all 8-minute rounds in a 6-hour period for the four classes of network behaviors identified in Table 2.

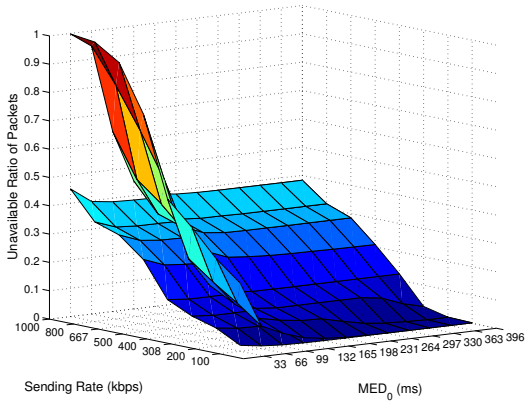


Figure 5: UPR is reduced when we add 100 ms buffering time to the predefined MED of an existing system. Top: UPR_0 at MED_0. Bottom: UPR_1 at $MED_0 + 100$ ms.

powerful a tool as increasing MED. We like to increase MED in such a way that human cannot perceive the difference in interactivities. Since signal quality will either be improved or remains the same, the overall QoE will be improved or remains the same.

To illustrate this idea, let MED_0 and UPR_0 (*resp.* $MED_1 = MED_0 + 100$ ms and UPR_1) be the MED and UPR of an existing system (*resp.* the improved system). Figure 5 shows the resulting UPR_0 and UPR_1, where the lower plane shows the reduced UPR_1 under MED_1 and the upper plane shows UPR_0 under MED_0. The results clearly demonstrate the merit of using additional MED to reduce UPR. Next, we investigate the amount that MED_0 can be increased without affecting interactivity.

5. EXTENDING MED BY JND

In this section, we define JND on MED in video conferencing and find its extent by subjective tests.

5.1 Comparative Subjective Tests

One method for evaluating the QoE of a test sequence is to present the original and the test sequences to a human subject and ask the subject to give an absolute score on the test sequence. After performing the experiments by many subjects, the mean opinion score (MOS) is derived by averaging the scores. Another approach is to present two test sequences to subjects and to ask each to indicate whether the sequences are different. This relative comparison allows subjects to perceive small differences in the two sequences and requires less expert knowledge from the subjects. However, it cannot lead to an absolute score on a test sequence [20].

In our subjective tests, we aim to find a range of MEDs within which subjects cannot perceive any change in interactivity in the conversation. This requires perceiving changes in the MED of a test sequence relative to the original sequence but does not require any absolute evaluation of the test sequence. As a result, the approach based on relative subjective tests is more appropriate.

In a relative subjective test, a two-way video conference session A (see Figure 6) is presented to a test subject, who is asked to compare to B, an identical two-way session with a different MED. The test subject is assumed to be sitting next to the party on the left, while listening to the conversation between the two parties.

Assuming the MED in A is fixed, we increase the MED in B and ask the test subject to determine which of the two sessions has a longer MED. JND is the difference between the MED in A and the MED in B at which subjects start perceiving a difference between A and B. Accordingly, subjects will not be able to differentiate

Figure 6: A video-conference session used in subjective tests

between the MEDs of A and B. In this paper, we aim at finding the JND of an existing system and use it to extend the system's MED to within JND in order to conceal its losses, while not incurring significant perceptual difference in interactivities.

Following Sat and Wah's definition on JND [32], let p_0 be the faction of subjects who correctly identify which of the two sessions has a longer MED. We define the 75% JND of delay as follows.

Definition 1: The 75% JND of MED_A is the maximum $|MED_B - MED_A|$ where $p_0 \leq 0.75$.

Although both the 50% JND and the 75% JND are generally used in psychophysics, we choose the 75% JND because 50% is a random guess according to our definition, and only more than 50% correctness is significant for perceiving differences.

Definition 2: A is perceptually the same as B ($MED_B \approx MED_A$) if MED_B is within the 75% JND region of MED_A.

Axiom 1. Reflectivity: The MED of any session is within its own JND region, since both are perceptually the same. This allows us to omit the comparison of a session with itself.

Axiom 2. Symmetry: $\approx$ is symmetric, since $|MED_B - MED_A| = |MED_B - MED_A|$. This allows the sessions to be presented in any order.

Axiom 3. IID: The subjects have the same level of expertise, and their ability to discover the difference in MED is independent and identically distributed (IID). This allows us to get the statistics of responses by repeated tests using multiple subjects.

Corollary 1. Non-transitivity: $\approx$ is not transitive. That is, $A \approx B$ and $B \approx C$ does not imply $A \approx C$. Hence, it will be necessary to carry out all pairwise comparisons with respect to A in order to determine the JND of A.

In our subjective tests with respect to a given session A, we prepare our test sessions with different MEDs and present them to subjects in pairs in a random order (according to *Axiom 2*). The subjects are then asked to identify the session with a longer MED. Based on the test results, we find the JND of A by identifying the MED where $p_0 = 0.75$ (according to *Definition 1*). We then repeat the experiments using another session with a different MED.

This method is similar to that of using constant stimuli but with two differences. First, we ask subjects to identify the session with a longer MED, instead of asking them whether they perceive a difference in the sessions. We have done it this way because our focus is on interactivity instead of the difference in the audiovisual content. Second, we compute our result from the responses of multiple subjects, rather than from repeated tests given to a single subject. This is due to the definition on JND, which is based on responses from multiple subjects (according to *Axiom 3*).

5.2 Experimental Results

In our experiments, we increase MEDs every 34 ms (which is slightly larger than the 33 ms frame period) in the $[0, 136]$ ms interval on top of the original MEDs of 102 ms, 238 ms and 374 ms, respectively. Because the network delay is changed in both directions, the mutual silence is actually increased by 68 ms in each case. We let subjects compare each of the three sessions with other sessions in the interval (by *Corollary 1*), and calculate p_0 for each case. We invited eight subjects to participate, and the entire set of tests took about 40 minutes for every subject.

Table 3: Conversational scenarios used in our subjective tests

Scenario	Talk Segment (s)	Mutual Silence (s)	Turns in a Minute
Short & Fast	0.76	1.07	33
Long & Slow	3.00	1.99	12

A test session with a longer MED is generated from the original face-to-face session by repeating some frames in the silence period. Instead of freezing the video in extending the MED, we play the last few frames back and forth in real time to simulate a natural movement of a human subject in the extended MED [40]. For audio, we simply insert silence during this period. Note that it will be impossible to record two identical conversations with different MEDs, since they will have differences, however small, in their audiovisual effects when recorded separately.

Our experiments are intended to answer the following questions:

- Does JND change as MED is increased?
- Does JND change under losses and delay jitters?
- Does JND change with the type of conversation?

We conduct our experiments on lossless, lossy, and jittery traces, and under two conversational scenarios shown in Table 3.

a) *JND under lossless networks.* Based on a previous result on voice-only conversations [32], JND tends to be shorter in conversations with a high turn frequency. To find a tight bound on JND, we use the first conversation in Table 3.

Figure 7(a) shows that, when the base MED gets larger, JND also becomes larger. This observation is consistent with the Weber-Fechner law, but with a shift of about 50 ms. This shift may be due to the fact that the video frame rate in the tested systems is at most 30 FPS, and that the new session cannot be differentiated from the reference session when the increase in its MED is less than 33 ms. Based on the observations, we model the JND of delay by the following linear function:

$$JND = k \times MED + c,$$

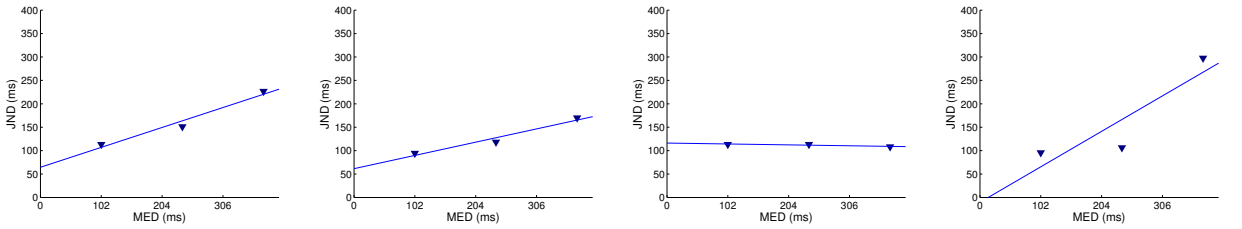
where k and c are constants for a given conversational scenario. The results show that JND is reasonably large even under a conversation with a high turn frequency.

b) *JND in lossy and jittery networks.* We have studied two representative traces using the first conversation in Table 3. The first trace is for a connection with 4.3% random losses, and the second is for a connection with 31 ms average delay jitters.

Figure 7(b) shows that in a lossy network, the relation between JND and MED is similar to that in a lossless network. However, Figure 7(c) shows that in a jittery network, JND appears to be larger when MED is small and is stable when MED is increased. This may be caused by the large number of late packets in a jittery network, leading to freezes in the audiovisual content and confusion of the subjects in identifying changes in MED.

c) *JND under a conversational scenario with a slow turn frequency.* We have conducted this experiment using the second conversation in Table 3. Figure 7(d) shows that JND is larger when MED increases, a result that is consistent with the observation in audio-only conferencing systems [32]. It also indicates that the turn frequency may change the sensitivity on delay, and that a slower conversation can tolerate a longer delay.

Our results show that JND is reasonably large under various network and conversational conditions. By increasing MED to within



(a) Lossless trace (b) Lossy trace (c) Jittery trace (d) Long & slow conversation

Figure 7: Changes in JND under different network and conversational conditions

JND, we can implement additional loss concealments for recovering lost or delayed packets, without affecting the interactivity of a conversation. In this way, the overall QoE can be improved.

6. ENHANCING EXISTING SYSTEMS

In this section, we present the loss-concealment methods made possible by the extended MED to within JND. We then demonstrate the approach for improving the QoE of existing systems.

6.1 Loss Concealments by Extending MED

In a jittery network, packets may arrive late under large jitters. In this case, the playout buffer will underflow, leading to freezes and degraded QoE. To smooth network jitters, a longer playout buffer can be used to store more packets and to smooth delay jitters.

On the other hand, in a lossy network, packets may be dropped and media data lost, again leading to freezes and low QoE. To recover those lost packets in real time, FEC can be used to add redundancy in transmission and to allow lost packets to be reconstructed. This approach will require additional bandwidth for sending redundant data and a longer playout buffer for receiving all the packets in an FEC block before carrying out recovery.

The size of the playout buffer (and MED) can be increased by JND without being perceptible. With this extra time, we have more room to smooth out jitters and recover lost packets. With MED increased by JND, the probability for a packet arriving late becomes

$$p_{\text{late}} = 1 - CDF(MED + JND(MED)),$$

where CDF is the cumulative distribution function of network delays. The probability that a packet is lost becomes

$$p_{\text{lost}} = \sum_{l=0}^{N_S-1} Pr(\text{only } l \text{ packets are received}),$$

where N_S is the number of source packets in an FEC block. The minimal buffering time for packets in an FEC block is

$$t_{\text{buffer}} = (N_S + N_R - 1)t_{\text{interval}},$$

where N_R is the number of redundant packets in an FEC block, and t_{interval} is the packet transmission period. With the additional JND, the number of redundant packets that can be used in FEC is

$$t_{\text{buffer}} \leq MED + JND(MED) - t_{\text{delay}},$$

where t_{delay} is a random variable of packet delay.

6.2 Design of a Packet Interceptor

In this section, we present the design of a packet interceptor to capture and modify the packet traffic in existing video conferencing systems. Our approach has two advantages. First, it can enhance an existing system and allows its performance before and after to be fairly compared. Second, it can be readily accepted by users of existing systems because it improves their perceptual quality without changing to a new system.

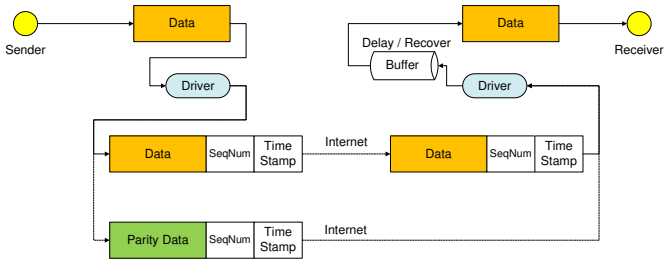


Figure 8: The interceptor deployed in Windows to intercept, modify and inject UDP traffic for proprietary systems.

We have picked Skype and MSN as our targets for improvement. By implementing the scheme in Windows kernel mode, it allows their traffic to be modified outside of their black-box designs. To do this in real time, we have developed a kernel driver using the Windows Filtering Platform [27]. This is part of the Windows Driver Kit in Windows 7 that provides ways to intercept, modify, and inject traffic in various layers.

In our implementation, we only intercept UDP traffic of the video conferencing system. As depicted in Figure 8, we add a sequence number and a time stamp to each packet before it is sent. In the receiver, the driver buffers the packets and releases each to the video conferencing system after reaching certain delay from the time each was sent. The additional data will not exceed the maximum transmission unit (MTU) of 1500 bytes, as the size of the largest packet produced by Skype is 1406 bytes and that by MSN is 1078 bytes.

For FEC protection, we send an extra packet with parity data for every several original packets. The parity packet is coded by Reed-Solomon code that allows a lost packet to be recovered if there is only one packet lost in the FEC block. After detecting a discontinuity in the sequence number, we recover the lost packet and send it to the video conferencing system. By succinctly choosing the size of the FEC block, the additional bandwidth incurred is small, while offering protection to the original audiovisual data.

6.3 RealTalk: A Testbed for Evaluating QoE in Proprietary Systems

Figure 9 shows the architecture of *RealTalk*, a testbed for evaluating proprietary video conferencing systems. The testbed consists of two Windows 7 machines serving as the video conferencing clients. The clocks of these machines are synchronized by Net Time Protocol (NTP) to ensure accurate measurements of conversational delays. An additional Linux machine serving as a network

Figure 9: The architecture of *RealTalk*, an evaluation testbed for proprietary video conferencing systems.

Table 4: QoE results of our proposed scheme

System	Interceptor	VQM	PESQ	MED (ms)	CS	CE	% of Subjects Preferring Scheme
Skype	Off	0.54	3.77	239	1.54	0.88	0%
	On	0.36	3.36	251	1.57	0.88	100%
MSN	Off	0.76	3.08	276	1.63	0.87	0%
	On	0.41	3.72	363	1.82	0.83	100%

emulator is connected to the two clients, where Trace Control for Netem [24], a trace-based network emulator, was installed to emulate different network conditions using the traces collected.

To generate input video frames in real time under various frame sizes and rates, we have developed a virtual camera program with Microsoft DirectShow. Audio is injected by Virtual Audio Cable [29], a software for redirecting audio from one source to another.

In our testbed, each client behaves like a human that speaks and replies using a pre-recorded audiovisual sequence. After detecting the end of the other party's speech segment (using special markers inserted into the audiovisual stream to indicate the start and end times of each speech segment), it waits for HRD before playing the next segment. This approach allows our testbed to simulate conversations with different delays from a single audiovisual source.

6.4 Experimental Results

We have found that Skype performs poorly under a jittery connection. Based on the 100-ms JND found in Section 5.2 for jittery connections and conversations with a high turn frequency (by interpolating the measurement results in the subjective tests), our traffic interceptor adds a 100-ms buffer to Skype to smooth its jitters. To ensure the same sending rate, we enforce the maximum bandwidth in the TCN network emulator. (Otherwise, the sender client in Skype will increase its sending rate after the jitters have been removed by our interceptor.) One way to limit the bandwidth without the TCN is to include another interceptor at the sender client.

Table 4 shows significant improvements in video quality with the use of our interceptor, where VQM has improved by 33%. Audio quality is decreased slightly by 11%, without being perceived in subjective tests. The degraded audio quality may be a result of the higher video quality. It is surprising to find that the overall MED increases by only 12 ms with 100 ms buffers inserted by our interceptor. This nonlinear change in MED may be caused by some time-consuming loss-concealment functions in Skype, which are bypassed when Skype finds a lower UPR. Figure 10(a) further illustrates the network condition used for testing Skype. The new operating point has significantly lower UPR, leading to better signal quality and a non-perceptible change in interactivity. Note that the reduced UPR is a result of the extended MED for smoothing delay jitters or performing FEC. Without the interceptor, there is no time to implement these loss-concealment schemes.

For MSN, we found that it performed poorly under lossy conditions. With a lossy connection and a high turn frequency, the JND is 138 ms (also obtained in Section 5.2). We thus add a 138-ms buffer in our interceptor to conceal those lost packets using FEC. Table 4 shows that, under the same network setting, video quality is improved by 46% and the audio quality is improved by 21%. The final MED is increased by 87 ms, which is within the target JND. For a similar reason as that in Skype, the better signal quality is due to the significant reduction in UPR (Figure 10(b)).

The last column of Table 4 also shows the results of the subjective tests conducted to determine the quality of Skype and MSN before and after deploying the interceptor. A total of 8 subjects were invited to perform the tests, and all found that the quality to be better with the interceptor included. Another experiment on an older version of Google Video Conferencing in Gmail [13] also showed the improvement of using our proposed interceptor in lossy networks. However, Google has recently changed its design from a peer-to-peer architecture to a server-client architecture, and it is not possible to use *RealTalk* to replay traces with this new architecture.

Our results clearly show that our scheme can improve the QoE of existing video conferencing systems, without incurring perceptible degradations on interactivity.

7. CONCLUSIONS

In this paper, we have proposed a novel method for improving existing video conferencing systems, based on observations of the current Internet behavior and human's ability to perceive changes in delay. Our work has focused on increasing MED, without incurring perceptible degradations on interactivity. By increasing MED by JND, existing systems have more room for buffering packets and for concealing losses, thus providing better signal quality without sacrificing interactivity. We have validated this approach by using a traffic interceptor added to Skype and Windows Live Messenger and have found significant improvements in QoE.

In the future, we plan to study trade-offs between improvements on signal quality and perceptible degradations on interactivity. With proper trade-offs, we expect to find operating points with better QoE when compared to those found in this paper. We also plan to study the run-time monitoring of network behaviors, which will allow JND to be determined dynamically in real time.

8. ACKNOWLEDGMENTS

We like to acknowledge the eight participants for helping us with the subjective tests, Wee-Hong Yeo for providing us with the raw video for subjective tests, and the staff in Microsoft for advising us on the Windows Filtering Platform. We truly appreciate the constructive comments and suggestions made by the reviewers.

9. REFERENCES

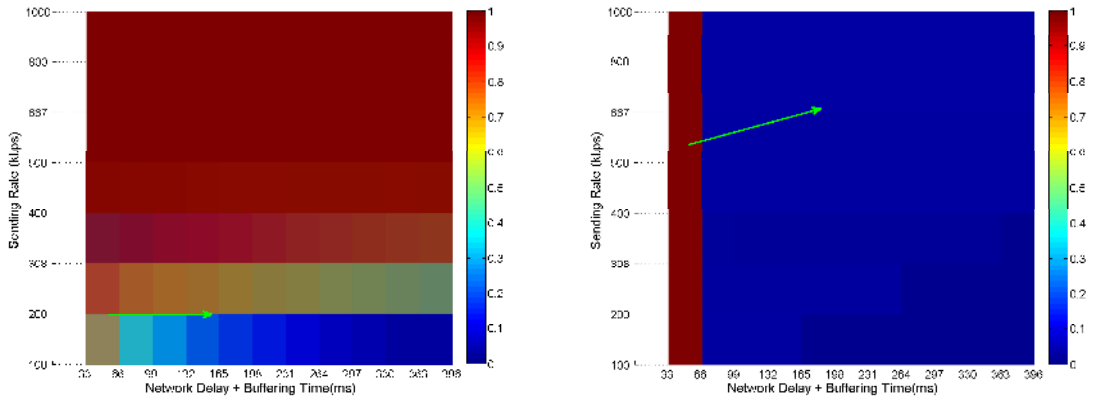
[1] B. Adelstein, T. Lee, and S. Ellis. Head tracking latency in virtual environments: psychophysics and a model. *Proc. of Human Factors and Ergonomics Society Annual Meeting*, 47(20):2083–2087, 2003.

[2] R. Barbosa, C. Kamienski, D. Mariz, A. Callado, S. Fernandes, and D. Sadok. Performance evaluation of P2P VoIP application. In *ACM NOSSDAV*, volume 7, 2007.

[3] S. Baset and H. Schulzrinne. An analysis of the Skype peer-to-peer Internet telephony protocol. *Arxiv Preprint CS/0412017*, 2004.

[4] J. Beerends, A. Hekstra, A. Rix, and M. Hollier. Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment. Part II: Psychoacoustic model. *J. of Audio Engineering Society*, 50(10):765–778, 2002.

[5] P. Biondi and F. Desclaux. Silver needle in the Skype. *Black Hat Europe*, 2006.



(a) Skype under a jittery connection

(b) MSN under a lossy connection

Figure 10: The change in UPR by using our traffic interceptor, where the green arrow indicates the change in the operating points. In MSN, the packet sending rate is increased by 20% in order to implement FEC, although the increased sending rate does not change the average loss rate and average jitter.

[6] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi. Detailed analysis of Skype traffic. *IEEE Trans. on Multimedia*, 11(1):117–127, 2009.

[7] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. Revealing Skype traffic: when randomness plays with you. *ACM SIGCOMM Computer Communication Review*, 37(4):37–48, 2007.

[8] O. Boyaci, A. Forte, and H. Schulzrinne. Performance of video-chat applications under congestion. In *Proc. 11th Int'l Symp. on Multimedia*, pages 213–218. IEEE, 2009.

[9] P. Calyam, M. Haffner, E. Ekici, and C. Lee. Measuring interaction QoE in Internet videoconferencing. *Real-Time Mobile Multimedia Services*, pages 14–25, 2007.

[10] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[11] L. De Cicco and S. Mascolo. A mathematical model of the Skype VoIP congestion control algorithm. *IEEE Trans. on Automatic Control*, 55(3):790–795, 2010.

[12] L. De Cicco, S. Mascolo, and V. Palmisano. Skype video responsiveness to bandwidth variations. In *Proc. of 18th Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 81–86. ACM, 2008.

[13] Google. Googla Talk Plug-in version 3.2.4.8431. http://mail.google.com.

[14] T. Hayashi, K. Yamagishi, T. Tominaga, and A. Takahashi. Multimedia quality integration function for videophone services. In *Global Telecommunications Conf., 2007. GLOBECOM'07. IEEE*, pages 2735–2739. IEEE, 2007.

[15] T. Hoßfeld and A. Binzenhöfer. Analysis of Skype VoIP traffic in umts: End-to-end qos and qoe measurements. *Computer Networks*, 52(3):650–666, 2008.

[16] T. Huang, K. Chen, and P. Huang. Tuning Skype's redundancy control algorithm for user satisfaction. In *Proc. INFOCOM*, pages 1179–1187. IEEE, 2009.

[17] T. Huang, P. Huang, K. Chen, and P. Wang. Could Skype be more satisfying? A QoE-centric study of the FEC mechanism in an internet-scale VoIP system. *Network*, 24(2):42–48, 2010.

[18] Z. Huang and K. Nahrstedt. Perception-based playout scheduling for high-quality real-time interactive multimedia. In *Proc. INFOCOM*, pages 2786–2790. IEEE, 2012.

[19] ITU. One-way transmission time. *Recommendation G.114*, 1996.

[20] ITU. Methodology for the subjective assessment of the quality of television pictures. *Recommendation I*, 2002.

[21] ITU. The E-model, a computational model for use in transmission planning. *Recommendation G.107*, 2005.

[22] S. Jelassi, G. Rubino, H. Melvin, H. Youssef, and G. Pujolle. Quality of experience of VoIP service: A survey of assessment approaches and open issues. *IEEE Communications Surveys Tutorials*, 14(2):491 –513, 2012.

[23] Y. Jia, W. Lin, and A. Kassim. Estimating just-noticeable distortion for video. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(7):820–829, 2006.

[24] A. Keller. Trace control for Netem. http://tcn.hypert.net/.

[25] W. Kho, S. Baset, and H. Schulzrinne. Skype relay calls: measurements and experiments. In *Proc. INFOCOM Workshops*, pages 1–6. IEEE, 2008.

[26] Y. Lu, Y. Zhao, F. Kuipers, and P. Van Mieghem. Measurement study of multi-party video conferencing. *Networking*, pages 96–108, 2010.

[27] Microsoft. Windows Filtering Platform. http://msdn.microsoft.com/en-us/library/windows/hardware/gg463267.aspx.

[28] Microsoft. Windows Live Messenger 15.4.3555.308. http://messenger.live.com.

[29] E. Muzychenko. Virtual Audio Cable 4.12. http://software.muzychenko.net/eng/vac.htm.

[30] M. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Trans. on Broadcasting*, 50(3):312–322, 2004.

[31] B. Sat and B. Wah. Analyzing voice quality in popular VoIP applications. *Multimedia, IEEE*, 16(1):46–59, 2009.

[32] B. Sat and B. Wah. Statistical scheduling of offline comparative subjective evaluations for real-time multimedia. *IEEE Trans. on Multimedia*, 11(6):1114–1130, 2009.

[33] Skype. Skype 5.10.0.116 Windows Edition. http://www.skype.com.

[34] K. Suh, D. Figueiredo, J. Kurose, and D. Towsley. Characterizing and detecting relayed traffic: A case study using Skype. In *Proc. IEEE INFOCOM*, volume 6, 2006.

[35] A. Takahashi, D. Hands, and V. Barriac. Standardization activities in the ITU for a QoE assessment of IPTV. *Communications Magazine, IEEE*, 46(2):78–84, 2008.

[36] A. Watson. Towards a visual quality metric for digital video. In *Proc. European Signal Processing Conf.*, volume 2, 1998.

[37] A. Watson and L. Kreslake. Measurement of visual impairment scales for digital video. In *Proc. of SPIE*, volume 4299, pages 79–89, 2001.

[38] Y. Xu, C. Yu, J. Li, H. Hu, Y. Liu, and Y. Wang. Measurement study of commercial video conferencing systems. Technical report, Polytechnic Institute of NYU, 2010.

[39] J. Xue and C. Chen. Mobile JND: environment adapted perceptual model and mobile video quality enhancement. In *Proc. of 3rd Multimedia Systems Conf.*, pages 173–183. ACM, 2012.

[40] W. H. Yeo. *Finding Perceptually Optimal Operating Points of a Real Tme Interactive Video-Conferencing System*. M.Sc. Thesis, Dept. of Electrical and Computer Engineering, Univ. of Illinois, Urbana, IL, May 2011.

[41] X. Zhang, Y. Xu, H. Hu, Y. Liu, Z. Guo, and Y. Wang. Profiling Skype video calls: Rate control and video quality. In *Proc. IEEE INFOCOM*, pages 621 –629, march 2012.