Quantifying QoS Requirements of Network Services: A Cheat-Proof Framework

Kuan-Ta Chen[1], Chen-Chi Wu[2], Yu-Chun Chang[12], and Chin-Laung Lei[2]

[1]Institute of Information Science, Academia Sinica
[2]Department of Electrical Engineering, National Taiwan University

ABSTRACT

Despite all the efforts devoted to improving the QoS of networked multimedia services, the baseline for such improvements has yet to be defined. In other words, although it is well recognized that better network conditions generally yield better service quality, the exact minimum level of network QoS required to ensure satisfactory user experience remains an open question.

In this paper, we propose a general, cheat-proof framework that enables researchers to systematically quantify the minimum QoS needs for real-time networked multimedia services. Our framework has two major features: 1) it measures the quality of a service that users find intolerable by intuitive responses and therefore reduces the burden on experiment participants; and 2) it is cheat-proof because it supports systematic verification of the participants' inputs. Via a pilot study involving 38 participants, we verify the efficacy of our framework by proving that even inexperienced participants can easily produce consistent judgments. In addition, by cross-application and cross-service comparative analysis, we demonstrate the usefulness of the derived QoS thresholds. Such knowledge will serve important reference in the evaluation of competitive applications, application recommendation, network planning, and resource arbitration.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*; H.4.3 [**Information Systems Applications**]: Communications Applications—*Computer conferencing, teleconferencing, and videoconferencing*

General Terms

Human Factors, Measurement, Performance

MMSys'11, February 23–25, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0517-4/11/02 ...$10.00.

Keywords

Psychophysics, Method of Limits, Quality of Experience, Crowdsourcing, Comparative Analysis

1. INTRODUCTION

Currently, it is not possible to guarantee the quality of packet delivery over the Internet because of unavoidable network impairment events, such as queuing delay, packet dropping, and packet reordering. Despite the Internet's unpredictable nature, researchers endeavor to design networked multimedia services that can always provide satisfactory user experience whatever level of network QoS (Quality of Service)[1] is provided. The research efforts sit from the core to the end-points, and address issues ranging from the packet level to the media level. For example, on the policy level, IntServ [5] attempts to guarantee the performance of mission-critical and real-time services, and DiffServ [2] is designed to provide differentiated quality levels for different needs. At the network layer, a number of router scheduling algorithms and their variants [20] try to provide prioritized and fair packet delivery. Meanwhile, at the application layer, numerous studies focus on the control of source traffic [10,27], the adjustment of VoIP playout buffer [21,29,32], and the concealment of information loss [13, 14, 28] to cope with the chaotic network impairments and provide the best possible service quality at their best.

Despite all the efforts devoted to improving the QoS of networked multimedia services, the *baseline* for such improvements has yet to be clearly defined. In other words, although it is widely recognized that better network conditions generally yield better service quality, *the exact minimum level of network QoS required to ensure user satisfaction remains an open question.*

For example, if network bandwidth is "insufficient," the quality of a VoIP call would degrade and the call parties would experience lower satisfaction. Normally a VoIP call requires tens of Kbps of bandwidth for the transmission of audio packets. However, the minimum bandwidth required by a particular VoIP service to barely ensure an "acceptable" degree of user satisfaction is undefined. This is also true in network gaming. Although short network delay is critical for interactive gaming, the exact minimum requirement of such delay (i.e., the longest acceptable delay) for gaming is unknown. In addition, the difference between the longest

[1] Here by network QoS we refer to the service level of network data delivery, including network bandwidth, network delay, packet loss rate, to name a few.

acceptable delay for slow-paced games, e.g., RPG (Role-Playing Games), and that for fast-paced gaming, e.g., FPS (First-Person Shooter) games, has yet to be determined.

We define the "minimum QoS needs" of a networked multimedia service as the minimum level of QoS that will yield an acceptable level of service quality from the user's perspective. In this case, users may feel that, although the service quality is far from perfect, it is acceptable so they will continue using the service. Quantifying the minimum QoS requirements for networked multimedia services is essential for the following tasks:

- **Network planning**: If the minimum QoS requirements for a networked multimedia service can be determined, networks can be planned accordingly. For example, if the longest acceptable delay for FPS games is 200 ms, an FPS game provider can ensure that game play is at least tolerable by hosting its game servers within 200 ms network delay from the majority of the players' locations.
- **Resource arbitration**: When network resources are limited, arbitration between different needs is necessary to avoid congestion collapse and subsequent service quality degradation. For example, suppose that the minimum bandwidth required for conference calls is 80 Kbps. To guarantee the quality of such calls, we can allocate at least 80 Kbps of bandwidth to each call, as long as other simultaneous needs do not have stricter real-time requirements.

We define an "intolerance threshold" as *the minimum level of a QoS factor that yields acceptable service quality from the user's perspective*. By definition, this threshold is not derivable by pure mathematical axioms and deductions; rather, it has to be extracted from users' opinions. The *de facto* subjective method for measuring the quality of a networked multimedia service is the MOS (Mean Opinion Score) rating test [15]. In an MOS test, subjects are asked to rate the quality level from Bad (the worst) to Excellent (the best), and the overall rating is obtained by averaging the scores from repeated tests. MOS scoring is widely used because it is simple and intuitive; however, it is not suitable for detecting the intolerance threshold for the following reasons.

1. The rating standard is somewhat obscure to experiment participants. As the concepts of the five scales, i.e., Bad, Poor, Fair, Good, and Excellent, cannot be concretely defined and explained, subjects may be confused about which scale they should give in each test.
2. In MOS tests, participants are asked to grade the MOS scores for stimuli; however, we do not know whether they pay full attention to the scoring procedures, or whether they just give ratings in a perfunctory manner. There is no established methodology for verifying the authenticity of a participant's ratings, and the measurement accuracy may be degraded due to untrustworthy participants [12].
3. Since the range of an MOS score is from 1 to 5, it is difficult to define an appropriate threshold that represents "the barely acceptable user experience." It may seem reasonable to take either 2 (Poor) or 3 (Fair) as the threshold. In VoIP quality tests, an MOS score of 4.0 is usually considered as the toll quality, but it does not represent the threshold for the barely tolerable service quality.

In this paper, using a psychophysical approach [30], *we propose a general, cheat-proof framework that enables researchers to systematically quantify the minimum QoS requirements for real-time networked multimedia services.* The framework not only enables us to measure the intolerance thresholds for QoS factors of interest, but also addresses the disadvantages of the MOS rating test mentioned earlier. In our experiments, a participant is simply asked to use the networked multimedia service under investigation. We adjust the service quality systematically over time, and the user clicks a dedicated button whenever he feels that the quality is intolerable. Obviously, the decision-making process here is simpler than that in the MOS test, since the five-scale rating is reduced to a dichotomous choice (i.e., whether or not the current service quality is acceptable). The features of the proposed framework are as follows:

1. It is *generalizable* across a variety of networked multimedia services. Thus, it can be applied to compare the resource demands of various services and a service's different implementations.
2. The participants do not have to describe the intensity of their sensations on a categorical or numerical scale. They only need to decide whether or not the current service quality is acceptable; thus, the burden on participants is much less than in the MOS rating experiments.
3. The framework is cheat-proof in that *the experiment results can be verified*. The verification relies on the consistency of each participant's inputs; that is, the service quality that a participant finds intolerable should be at similar levels in repeated tests. By employing this property, we can detect inconsistent judgments and remove problematic data before performing further analysis and modeling.

To evaluate the proposed framework, we conducted a pilot study that targeted three real-time networked multimedia services, namely, VoIP, video conferencing, and network gaming, including six applications that provide those services. In the study, the minimum network bandwidth, as well as the maximum packet loss rate and network delay, for the applications were assessed based on $1,037$ experiments involving 38 participants and $13,184$ click actions. The results show that the judgments made by different participants were highly consistent with one another, which confirms the reliability of our framework and validates the derived QoS needs of networked multimedia services. We also provide cross-application and cross-service comparative analyses and discuss their implications.

Our contribution in this work is three-fold:

1. We propose a general, cheat-proof framework for quantifying the minimum QoS needs of real-time networked multimedia services. The most important features of the framework are that i) the experiment procedure is simple, so even inexperienced participants can make consistent judgments easily; ii) it enables us to employ crowdsourcing strategy because it supports systematic verification of the participants' inputs [3, 31]; and iii) its measures the quality of a service that users find

intolerable in a natural way instead of relying on artificial thresholds.

2. The framework enables *cross-application comparative analysis* of applications' minimum network QoS needs. Therefore, it can be used to compare the design and implementation of an application with competing applications in terms of their resource demands. It can also be used to recommend the most suitable networked multimedia application to end-users based on the capacity and congestion level of their access networks (cf. Section 4.3).
3. The framework also allows us to perform *cross-service comparative analysis* of networked multimedia services' resource demands. Thus, it can be used to quantify the intrinsic discrepancy of QoS needs between different services, e.g., between VoIP and video conferencing (cf. Section 4.4). Moveover, the quantification results provide information that is essential to network planning and resource arbitration for the provision of quality services.

The remainder of this paper is organized as follows. Section 2 contains a review of related works. We elaborate on our proposed framework in Section 3. In Section 4, we discuss the pilot study conducted on three real-time networked multimedia services to validate the framework's ability to derive minimum QoS needs and demonstrate its use. Finally, in Section 6, we present our conclusions and consider future research directions.

2. RELATED WORK

Although a great deal of effort has been devoted to improving the quality of networked multimedia services, relatively little research has been done to understand the minimum QoS needs of such services. According to [17] and the ITU-T E-model [16], the maximum allowable end-to-end delay for a satisfactory VoIP conversation is 150 ms, but it is not clear how this value was derived. While the subjective experiments for constructing the E-Model training data were based on the MOS rating test, the threshold is specified by setting a certain MOS score as the intolerance threshold, which may not faithfully reflect users' intolerance levels.

In [23], it is suggested that the intolerable packet loss rate should be 1% for high-quality audio-video streaming, and 2–3% for two-way interactive conferencing based on the recommendations of the Study Group 12 of ITU-T. Once again, how these thresholds were derived is not reported. Moreover, the thresholds ignore the discrepancies between applications, each of which may have a distinct intolerable loss rate due to different codec choices and data transmission strategies (which we will show by experiments in Section 4.3). Therefore, the applicability of the thresholds is questionable.

In [4], Bouch et al. proposed an experiment design for assessing the minimum QoS needs of network audio applications. In the experiments, two participants were asked to play a word-guessing game where they could only communicate with each other via VoIP and adjust the network quality by using a software slider at the same time. The participants were expected to find the lowest network quality that provided the least acceptable voice quality for game play, but there was no mechanism for validating whether participants followed the guidelines. Hence, careless or untrustworthy participants could skew the experiment results by, for example, focusing on the word-guessing game and randomly dragging the slider backwards and forwards.

A few studies have proposed to adopt the psychophysical approach [30] to understand the acceptability of certain multimedia content in terms of users' perceptions [1, 19, 22]. For instance, in [22], McCarthy et al. asked participants to watch 210-second test clips in which the video quality is increased or decreased every 30 seconds and report whenever they feel the quality acceptable or unacceptable. The authors varied the degree of quantization and/or the frame rate, and measured the perceived quality by calculating the ratio of time the video quality is acceptable by users. The experiment results indicated that users prefer high resolution over high frame rates.

This work also adopts the psychophysical approach; however, it differs significantly from previous studies (e.g., [1, 19, 22]) in a number of ways:

1. Rather than using the traditional "Method of Limits" test [30] for merely a study on the factors that may affect multimedia content quality, we extend the test with a more careful control of factor magnitude (c.f., Section 3.1) and a cheat proof mechanism (c.f., Section 3.2). We intend to make the proposed framework as general as possible so that researchers and practitioners can base on the framework for further studies.
2. We focus on the minimum acceptable level of QoS that should be provided for a networked multimedia service, rather than on the quality of source multimedia content.
3. Our framework is unique in that it supports the verification of users' inputs, which may be untrustworthy if a user experiment is outsourced or even crowdsourced [9, 11][2]. We believe this feature makes the proposed framework particularly useful as crowdsourcing is now gradually adopted in the research community [24].

3. THE PROPOSED FRAMEWORK

In this section, we describe our framework for assessing the intolerance thresholds of network QoS factors from the user's perspective. First, we consider the design of the experiments in which participants are asked to click a dedicated button whenever the service quality becomes intolerable. Second, we explain how we identify inconsistent judgments provided by malicious or perfunctory participants. We conclude this section with the derivation of the intolerance threshold of the QoS factor of interest.

In our framework, each experiment configuration focuses on one QoS factor for a service. Hereafter, we refer to the target networked multimedia service as "the service," and the target network QoS factor as "the QoS factor."

3.1 Experiment Design

Our framework basically adopts and extends the "Method of Limits" approach from Psychophysics [30] by a more careful control of QoS factors and cheat proof support. In our

[2] A cheat-proof framework for QoE evaluation has been proposed in [9] and [8], but it is targeted at a different goal, that is, to quantify the QoE of multimedia content, rather than to find the minimum QoE levels of network systems.

experiment, we systematically alter the quality of the networked multimedia service by controlling the QoS factor while the participants use the service. Participants are asked to press a dedicated button whenever they feel that the degradation in quality is unacceptable. This design is simple and intuitive in that the participants do not need to be well-trained to make a simple dichotomous decision (i.e., decide whether the current service quality is tolerable), and the click action is straightforward. When a participant clicks the button, we record the current magnitude of the QoS factor, and designate it as the intolerance threshold sample (ITS) generated by the click.

The rationale behind our experiment design is simple. Provided that a participant's click decisions are based purely on his perceptions of the service quality, and his intolerance threshold samples are self-consistent, the average of those samples can be treated as the intolerance threshold of the QoS factor from the participant's perspective. However, this raises two major issues in the experiment design: 1) *How should we ensure that a participant makes click decisions based purely on his perceptions*; i.e., how can we prevent a participant from "predicting" the magnitude of the QoS factor and making click decisions accordingly? 2) *How should we judge the consistency of a participant's intolerance threshold samples?* We discuss the first issue below and consider the second in Section 3.2.

During an experiment, we systematically vary the magnitude of the QoS factor, which determines the service quality, to "explore" a participant's intolerance threshold for that factor. Meanwhile, we have to ensure that the participant cannot predict the magnitude of the QoS factor; otherwise, he could report that the quality is intolerable based on timing predictions and still remain highly consistent with his intolerance threshold samples. An experiment is comprised of a number of cycles, each of which contains two stages, *the plateau stage* and *the probing stage*, and one operation called *quality boosting*. Basically, we maintain the service quality for an unspecified period (the plateau stage), and gradually degrade the service quality until the participant becomes intolerant of the quality and clicks the button (the probing stage). Then, we raise the service quality to a certain level (quality boosting) and proceed to the next cycle. Figure 1 illustrates the evolution of the service quality over time. Without loss of generality, we assume that the QoS factor and the service quality are positively correlated; that is, the higher the QoS factor, the better the service quality. If a QoS factor, e.g., the network loss rate, correlates negatively with the service quality, we simply reverse the direction of changes; in other words, to degrade the service quality, we tune the QoS factor higher rather than lower. Next, we consider the design of the plateau stage, the probing stage, and the quality boosting operation.

3.1.1 Plateau Stage

The plateau stage has two functions. One is to remind the participants how the reasonable service quality should be; thus, occasionally we need to reinforce the point by providing a reasonable service quality for a certain period. The second is to prevent the participants from predicting the degradation pattern of the service quality; therefore, the process of the quality degradation must include a certain amount of randomness. This explains why we use a variable-length plateau stage before the probing stage. To achieve a balance between the experiment's efficiency and the predictability of quality degradation, we choose a random period between 2 and 6 seconds for each plateau stage.

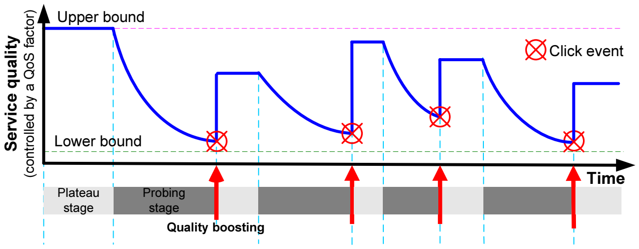

Figure 1: An evolution of the service quality over time. Each cycle starts with a plateau stage, which is followed by a probing stage, and terminates with a quality boosting.

3.1.2 Probing Stage

The probing stage is designed to discover a participant's intolerance threshold for a QoS factor by gradually degrading the service quality. The procedure needs to be planned carefully because there is a response delay between a participant's perception and his click action. The delay is unavoidable since a participant needs time to assess the current service quality and react to it accordingly. Suppose that a participant clicks the button at time t_x, his response delay is t_{delay}, and the magnitude of the QoS factor at time t is $Q(t)$. Then, we would obtain an intolerance threshold sample $Q(t_x)$. However, if we consider the response delay, the exact time that the participant feels intolerant should be $t_x - t_{delay}$, and the magnitude of the actual intolerance threshold sample should be $Q(t_x - t_{delay})$. Since the response delay of each click action is neither constant nor measurable, we can only compensate for it by reducing the difference between $Q(t_x)$ and $Q(t_x - t_{delay})$. In other words, to ensure that the measured intolerance threshold sample is close to the actual sample, the service quality should not degrade too rapidly. On the other hand, if the service quality degrades too slowly, it will elongate the experiment time and lower the data collection efficiency.

To achieve a balance between the experiment's efficiency and the accuracy of intolerance threshold samples, we devised a strategy that degrades the service quality at an inconstant rate. Specifically, the quality degradation in the experiments follows a parameterized exponential decay function. The unit exponential decay function is defined as

$$N(t) = N_0 e^{-\lambda t}, \tag{1}$$

where $N(t)$ denotes the quantity at time t, N_0 is the initial quantity, i.e., the quantity at time $t = 0$, and λ is the decay constant. The function describes how a variable declines from N_0 to 0 over time, where the rate of decline is decided by λ. We extend the function by adding three parameters: an upper bound, a lower bound, and the time allowed for decline. The extended function allows us to compute the magnitude of the QoS factor at time t by

$$Q(t) = Q_{lb} + e^{-\lambda t/T}(Q_{ub} - Q_{lb}), \tag{2}$$

where $Q(t)$ denotes the magnitude of the QoS factor at time t; Q_{ub} and Q_{lb} denote the upper bound and lower bound of

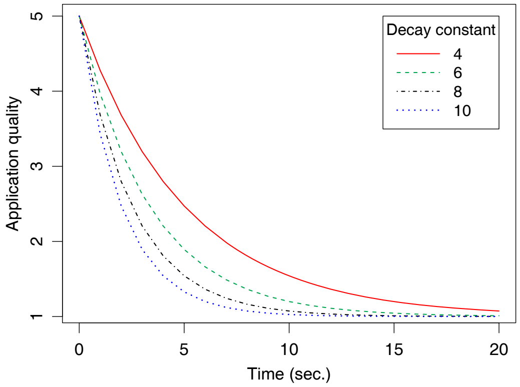

Figure 2: Exponential decay process of an application's quality with decay constants of 4, 6, 8, and 10 within a 20-second period

the QoS factor respectively; and T is the time required for the QoS factor to decline from Q_{ub} to Q_{lb}. Figure 2 illustrates the exponential decay process of the service quality, where the probing range of the QoS factor is between 1 and 5. The QoS factor decreases from the upper bound (5) to the lower bound (1) in 20 seconds, while λ equal 4, 6, 8, and 10 for the respective bounds. As shown on the graph, if a larger decay constant is used, the QoS factor will decline more rapidly toward the lower bound.

By adopting the exponential decay strategy, the service quality will decline rapidly initially, and slow down as the QoS factor approaches the lower bound. The upper bound and the lower bound of the QoS factor must be chosen carefully based on the guidelines: 1) the upper bound should yield a satisfactory service quality; and 2) the lower bound should yield a clearly intolerable service quality. A participant is more likely to feel dissatisfied with the quality when the QoS factor is closer to the lower bound; therefore, we can ensure the experiment's efficiency and the accuracy of the intolerance threshold samples by adopting a rapid decline followed by a gradual decline during the probing process. Based on our experience, setting λ between 2 and 6 is a reasonable choice.

The parameter T also needs to be randomized to reduce the predictability of the magnitude of the QoS factor. Considering the balance between the efficiency and the predictability, we set T uniformly distributed between 15 and 25 seconds. If the QoS factor reaches the lower bound, it will not change until the participant clicks the button. However, this is unlikely to happen if the lower bound is chosen properly, since the service quality at that point should be much worse than "barely acceptable."

3.1.3 Quality Boosting

Each time a participant clicks the button, a probing period terminates and a quality boosting operation is performed immediately; that is, the service quality is increased by a certain amount (i.e., by controlling the QoS factor). The mechanism serves two purposes. First, it provides prompt feedback on a participant's click action. This makes the action worthwhile because it seems to "rescue" the service from serious quality degradation. It also facilitates interaction and adds fun to the experiments. Second, since the quality boosting operation raises the QoS factor toward the upper bound, the experiment process can transit smoothly to the plateau stage in the next cycle.

The amount of quality boosting is not constant. We cannot always reset the QoS factor to its upper bound because a participant may get used to the pace of quality degradation and detect its regularity. Therefore, in each quality boosting operation, the QoS factor is raised to a random level between the current value and its upper bound. As long as the length of the plateau stage, the degradation rate of the service quality, and the amount of quality boosting are unpredictable, the participant has no choice but to pay full attention to the varying service quality if he is to make consistent judgments.

3.2 Cheat Proof Mechanism

In each experiment, we collect the intolerance threshold samples generated by the participants' click actions. If the click actions are made randomly, either unintentionally or mischievously, they will lead to inaccurate assessment of the intolerance threshold; therefore, methods must be devised to detect such inputs. In addition, punishment and reward rules can be set to encourage participants to provide quality judgments. For example, participants who provide problematic feedback do not get rewards.

Suppose a participant has made n click actions in an experiment and produced intolerance threshold samples $\mathbf{v} = (v_1, v_2, ..., v_n)$, where v_i denotes the magnitude of the QoS factor when he made the i-th click. We assume that v_i are independent and identically distributed random variables. This is reasonable because a participant's assessments of intolerable quality each time should be independent and his standard should not change over time. Thus, if we randomly split $\mathbf{v}$ into two disjoint sets $\mathbf{v_a}$ and $\mathbf{v_b}$, they are likely to have statistically identical distributions. Based on the rationale, we apply the Wilcoxon rank-sum test [31] to determine the consistency of a participant's intolerance threshold samples by testing if $\mathbf{v_a} \sim \mathbf{v_b}$ holds. Specifically, we randomly and divide $\mathbf{v}$ into two equal[3] disjoint sets $\mathbf{v_a}$ and $\mathbf{v_b}$. Then we perform a Wilcoxon rank-sum hypothesis test on the two subsets with the null hypothesis that the distributions of $\mathbf{v_a}$ and $\mathbf{v_b}$ are drawn from a single population. If the computed p-value is above the desired significance level, it means that we cannot reject the null hypothesis. In this case, we consider that the participant's intolerance threshold samples are consistent; otherwise, we deem that his judgments are not trustworthy.

As the above hypothesis test can be biased by how $\mathbf{v}$ is divided into two sets, we extend it to an m-fold version to enhance its robustness; that is, we perform the test m times. Since we perform a total of m hypothesis tests in parallel, the desired significance level in each fold must be corrected, or the overall confidence will be not equal the expected $(1 - \alpha)$, assuming that the desired significance level is α. Thus, we apply the Bonferroni method [3] so that the significant level in each fold is α/m. Then, if all the p-values from the m tests are above α/m, we conclude that the participant's judgments are self-consistent; otherwise they are considered inconsistent. From our data set, we find that setting m to 30 is sufficient to achieve a reliable result when testing the consistency of a participant's behavior.

[3] If the size of $\mathbf{v} = n$ is odd, we split $\mathbf{v}$ into two sets where the size of one set is $(n-1)/2$ and that of another is $(n+1)/2$.

3.3 Intolerance Threshold Estimation

Suppose that n_p participants have conducted a total of n_{exp} experiments ($n_{exp} \geq n_p$); for each participant i, we have collected a set of intolerance threshold samples $\mathbf{v}_i$. The estimation of the intolerance threshold begins by applying the behavior consistency test on $\mathbf{v}_i$, $1 \leq i \leq n_p$, and removing the participants whose behavior is not self-consistent. Then, for each of the remaining participants, we compute the average intolerance threshold as $IT_i = mean(\mathbf{v}_i)$, where $mean(\cdot)$ is the function of the arithmetic mean. Finally, we calculate the intolerance threshold of the QoS factor by taking the average of the intolerance thresholds of the behavior-consistent participants as $mean(\{IT_i\}), i \in$ the set of behavior-consistent participants.

4. PILOT STUDY

In this section, we present a pilot study of three real-time networked multimedia services based on the proposed framework. The purpose of the study is four-fold:

1. To show that even inexperienced participants can produce consistent judgments easily, i.e., intolerance threshold samples (ITS), if they stay focused on the experiments.
2. To show that the intolerance threshold samples of different participants are mutually consistent. This confirms the robustness of our framework and validates the derived minimum QoS needs of networked multimedia services.
3. To demonstrate that the framework facilitates comparisons of the QoS needs of *different implementations* that provide identical networked multimedia services. For example, we compare four VoIP products, namely, AIM, MSN Messenger, Skype, and Google Talk, in terms of their minimum QoS demands.
4. To demonstrate that the framework facilitates comparisons of the QoS needs of *different networked multimedia services*, and provide quantitative results that are essential to network planning and resource arbitration. For example, we compare the minimum bandwidth requirements of VoIP, video conferencing, and network gaming.

We begin with a description of the experiment setup and a summary of the collected data. Then, we verify the consistency of the measured intolerance threshold samples from several aspects, and examine the derived intolerance thresholds for the compared applications and services. Finally, based on the results, we perform cross-application and cross-service comparative analysis of their minimum QoS requirements.

4.1 Experiment Description

4.1.1 Studied Services and Applications

We consider three real-time networked multimedia services, namely, VoIP, video conferencing (referred to as conferencing hereafter), and network gaming. For VoIP, we select four popular applications for investigation: AOL Instant Messenger 6.9 (AIM), MSN Messenger 2008 build 8.5, Skype 3.8, and Google Talk 1.0. In addition, we conduct conferencing experiments on the first three applications because Google Talk does not support conferencing. For network gaming, we select two genres, FPS (First-Person Shooter) games and RPG (Role-Playing Games). We consider three games: Unreal Tournament 3 (UT), which is an FPS game; and Lineage 2 (Lineage) and World of Warcraft (WoW), which are RPGs. The selection of games enables us to perform both intra-genre and cross-genre analysis of the games' QoS demands.

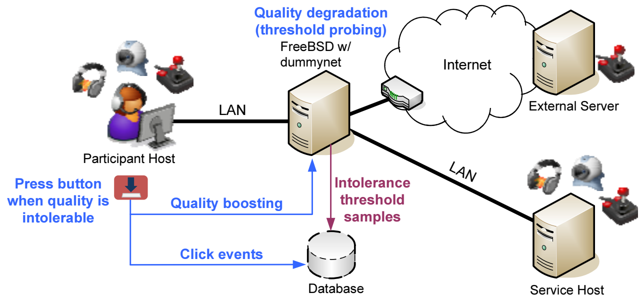

Figure 3: The environment setup for the VoIP, video conferencing, and network gaming experiments conducted in the pilot study

4.1.2 Environment Setup

We use three computers in a LAN to conduct the experiments for assessing the QoS needs of the three services. The first computer, the "participant host," is designated for the experiment participants; the second, the "service host," provides the content of the investigated service; and the third, the "router," connects the participant host and the service host to control the traffic between the two hosts. We run `dummynet` on the router to control the delay, bandwidth, and packet drop rate of the traffic passing through the router. Some applications have a centralized server-client architecture, so we have to connect to an external server instead of using our own service host. In these cases, the traffic from the participant host will flow through the router before reaching the external server, and vice versa. The setup of the network and facilities is illustrated in Figure 3.

In each VoIP experiment, we establish a VoIP call between the participant host and the service host. A two-minute song is played as the audio input on the service host, and the participants hear it from the participant host because of the VoIP connection. The participants are asked to listen to the VoIP-relayed song and click the button whenever the music quality (amount of degradation) becomes unacceptable. Similar settings are used in conferencing experiments, except that the song is replaced by a video clip, and the participants are asked to watch the network-relayed video (with sound tuned mute) instead of listening to music.

In the FPS game experiments, UT is configured in the Deathmatch mode, which means that each player must kill as many characters as possible, or his characters will be endangered. To induce more interaction, we put six bots in the game, so a participant will always encounter six opponents. In the RPG game experiments, external servers hosted by the game operators are used in place of the service host. Because the Internet induces additional latency, we compute the overall round-trip time (RTT) as the sum of the `dummynet`-injected RTT and the Internet RTT, which are derived from on the respective timestamps and sequence numbers of TCP packets. In the experiments, the partic-

Table 1: Summary of the experiment results in the pilot study

Service	QoS Factor	Application	# Users	# Exp.	# Clicks	Inter-click Time (secs)	Average ITS	95% Confidence Band of ITS	Lower Bound	Upper Bound
VoIP	Loss rate (%)	AIM	16	74	1, 059	8	9.2	(9.0, 9.4)	0	20
		MSN Messenger	15	69	824	9	10.8	(10.5, 11.1)	0	20
		Skype	15	66	898	8	8.2	(7.9, 8.5)	0	20
		Google Talk	15	62	985	7	8.1	(7.9, 8.3)	0	20
	Bandwidth (Kbps)	AIM	15	41	462	10	27.3	(26.4, 28.2)	10	80
		MSN Messenger	15	40	626	7	39.6	(38.5, 40.6)	10	80
		Skype	14	40	688	7	43.5	(42.6, 44.5)	10	80
		Google Talk	15	42	481	10	29.3	(28.1, 30.4)	10	80
Conferencing	Loss rate (%)	AIM	12	42	529	9	11.4	(11.1, 11.7)	0	20
		MSN Messenger	11	35	552	7	8.9	(8.6, 9.1)	0	20
		Skype	11	38	381	11	12.8	(12.4, 13.2)	0	20
	Bandwidth (Kbps)	AIM	11	36	413	10	60.5	(58.4, 62.6)	30	200
		MSN Messenger	11	43	490	10	80.6	(77.2, 84.0)	30	280
		Skype	11	33	302	12	78.9	(73.7, 84.2)	30	350
Gaming	RTT (sec)	Lineage	21	74	1, 080	19	0.77	(0.75, 0.79)	0	0.80
		WoW	19	68	681	27	0.93	(0.91, 0.96)	0	0.80
		UT	21	72	925	21	0.79	(0.77, 0.81)	0	0.80
	Bandwidth (Kbps)	Lineage	16	53	681	22	6.2	(6.0, 6.5)	1	15
		WoW	16	56	503	30	9.2	(8.9, 9.5)	5	25
		UT	16	53	624	23	16.9	(16.6, 17.1)	15	30
Overall			38	1, 037	13, 184	13				

ipants are asked to continuously interact with other characters and the environment, such as by fighting monsters, picking up items on ground, or interacting with NPCs (non-player characters), because network impairments only affect gaming experience during such interaction. In addition, to make the game play environment comparable, the participants were asked to operate their characters in a region where there were about 20 monsters and several NPCs that they could interact with.

4.1.3 Data Summary

We hired 38 part-time employees to conduct experiments on an overall of 20 service-application-QoS-factor configurations. The participants performed 1, 037 experiments and 13, 184 click actions, which took 47.6 hours in total. Table 1 summarizes the experiments and the collected intolerance threshold samples.

4.2 Consistency Checks

Next, we examine whether our framework yields consistent intolerance threshold estimates contributed by the experiment participants.

4.2.1 Consistency of Individual Participants

We begin by assessing the consistency of each participant's judgments (c.f., Section 3.2). The results show that 97% (245 out of 253) of experiment-participant pairs passed the test with a significance level of 0.05. We believe this rate satisfactory given that all the participants are regular computer users without specialized training in network or multimedia QoS. The only guideline given to the participants before the experiments began was "click the dedicated button whenever you find the service quality intolerable."

4.2.2 Consistency of Overall Inputs

Next, we examine the consistency of the overall intolerance threshold samples (ITS) that comprise the ITS contributed by all participants with the same experiment setting. Figure 4 shows the distributions of the overall ITS. Samples provided by a participant have a distinct color and mark. Clearly, the ITS for an application generally cluster around a certain value, even if they are from different participants. Moreover, the dispersion of the clusters varies across services and QoS factors. We believe that the variability in the overall ITS is caused by two factors. 1) Participants may use different criteria for an "acceptable quality." This explains the disagreements between the ITS from different participants. 2) Applying identical network impairments to a service does not necessarily lead to the same quality degradation because of *the variation in the service's workload*, mostly *the multimedia content*. For example, the impact of insufficient bandwidth on conferencing may vary because the service's bandwidth requirement is highly dependent on the complexity of the video frames, which may vary significantly over time. This property explains why the effect of bandwidth on VoIP/conferencing and the effect of network delay on gaming induce more variability in the overall ITS than the other experiments. Specifically, the impact of packet loss on VoIP/conferencing is relatively stable, since the router always drops a certain proportion of packets regardless of the service's current workload; however, this is not the case with insufficient network bandwidth. In addition, the games' bandwidth usage is more constant [7], so the impact of network bandwidth on gaming is relatively stable, as shown in Figure 4. On the other hand, the impact of delay on gaming is highly dependent on the workload, i.e., the action that a game character is performing or trying to perform.

4.2.3 Consistency across Participants

Our last check assesses the consistency of different participants' judgments. It is expected that the consistency between different participants would be lower than that of a single participant. However, it is a challenge to pinpoint what degree of consistency could be considered "reasonable." Since our experiments evaluate several applications for each service, and the applications' QoS needs are usually different, we adopt a "relative comparison" approach. That is, if the ITS of an application from several participants is more consistent than that of different applications from individual participants, we consider that the judgments on intolerable thresholds are consistent across the experiment participants.

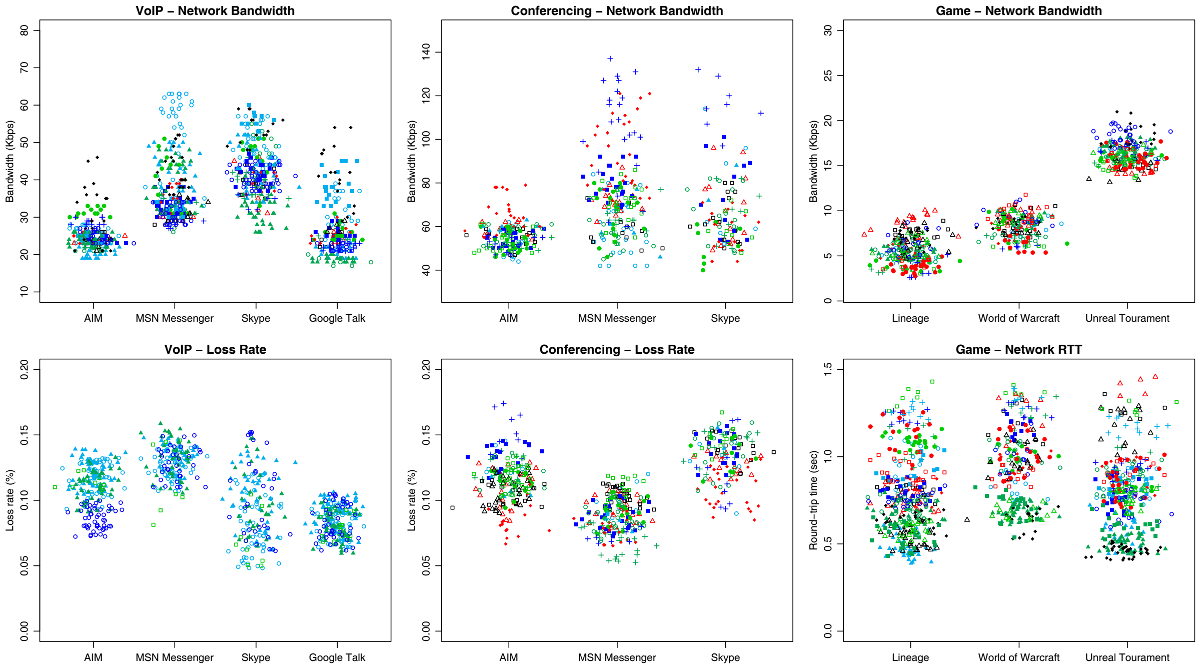

Figure 4: The scatter plots of the collected intolerance threshold samples in the pilot study

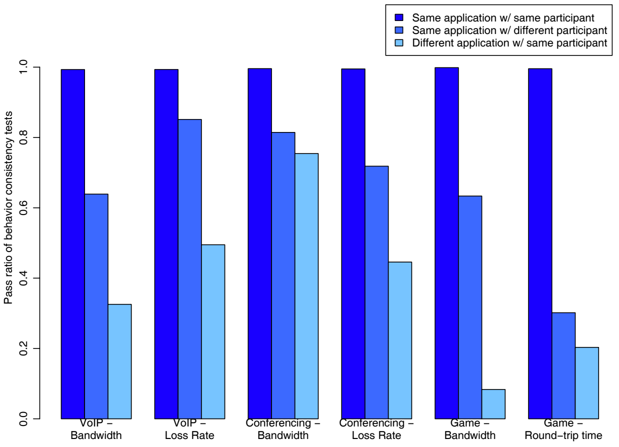

Figure 5: Behavior consistency tests across participants and applications

To perform this check, we apply the behavior consistency test for cheat proof (Section 3.2) in three cases: 1) the ITS of an application from a single participant; 2) the ITS of an application from different participants; and 3) the ITS of different applications from a single participant. Figure 5 shows the proportion of tests that passed the consistency test with a significant level of 0.05 in each case. On the graph, nearly all the pass ratios of case 1 reach 1 because the participants' judgments were generally consistent. We compare the pass ratios of case 2 and case 3, and find that, for each service-QoS-factor combination, the pass ratio of case 2 is always higher than that of case 3. The result indicates the robustness of our experiment design in that a group of untrained participants with different backgrounds can achieve a consensus about the minimum QoS needs of an application. Furthermore, it validates our derivation of the intolerance threshold by taking an average of the participants' judgments.

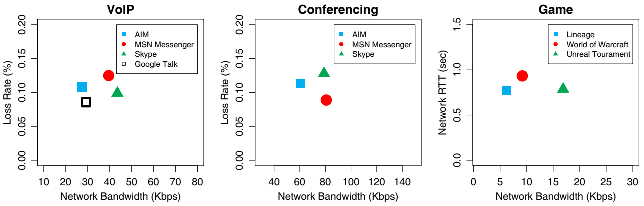

Figure 7: Comparison of the minimum QoS needs for applications that provide the same service

4.3 Intra-Service Application Assessment

In this subsection, we discuss the intolerance thresholds derived from the experiment results. Figure 6 shows the intolerance threshold for each network QoS factor in each application, where the vertical bars represent the factors' 95% confidence bands. The summarized version is shown in Figure 7, where the x-axis and y-axis represent the intolerance thresholds of the two QoS factors we examine for each networked multimedia service.

4.3.1 VoIP

Figure 6 shows that different VoIP applications have different requirements in terms of network bandwidth and packet loss rate, even though they all provide VoIP services. In terms of bandwidth, Skype is the most demanding because it requires at least 42 Kbps to ensure an acceptable voice quality. In contrast, 27 Kbps bandwidth is sufficient for Google

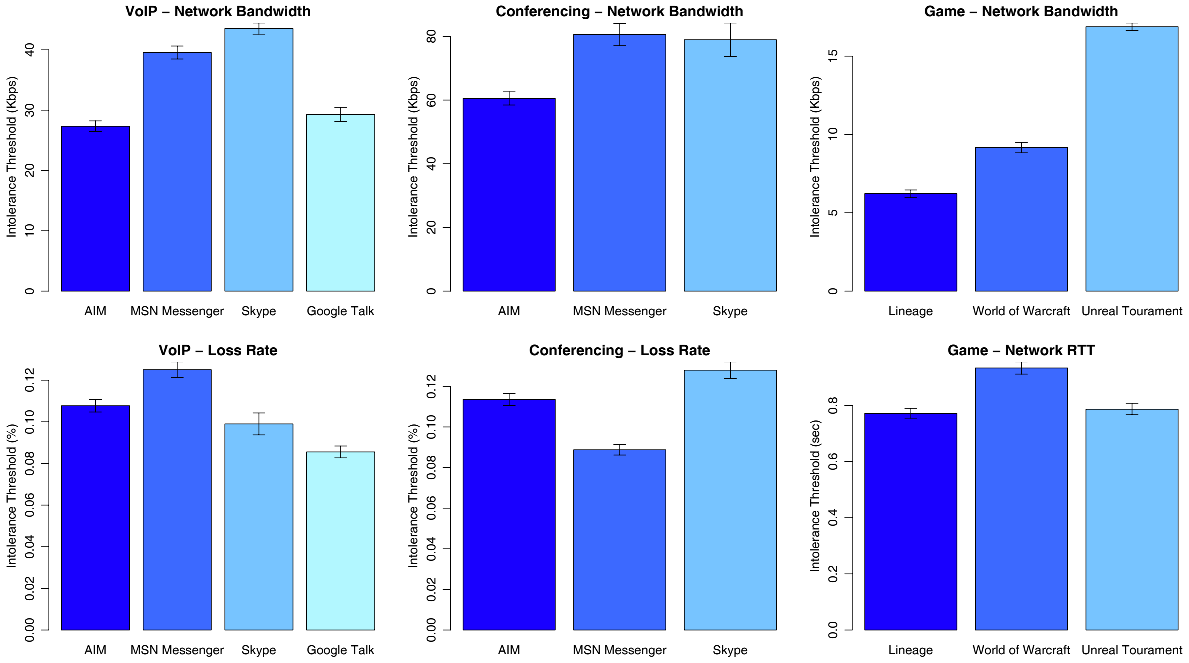

Figure 6: The derived intolerance thresholds for all the applications and services in the pilot study

Talk to provide a tolerable quality. In terms of packet loss rate, however, Google Talk is the most demanding application (8.5%), and MSN Messenger exhibits the strongest resilience against packet loss (12.5%).

We also compare the QoS needs of four VoIP applications when both QoS factors are considered, as shown in Figure 7. Among the four VoIP applications, MSN Messenger and Skype can be classified in one group, and AIM and Google Talk can be placed in another. The first group is more demanding in terms of network bandwidth, but relatively more robust in terms of packet loss. On the other hand, the second group requires less bandwidth to maintain an acceptable quality, but is relatively more sensitive to network loss. We believe that the chart provides useful information for network planning and application selection. For example, if network bandwidth is a concern, then AIM and Google Talk would be the preferred choices; otherwise, MSN Messenger and Skype may be considered because they are relatively more robust against packet loss.

4.3.2 *Video Conferencing*

According to Figure 6, MSN Messenger and Skype are similar in terms of their minimum bandwidth requirement (80 Kbps), while AIM's requirement is much lower (60 Kbps). On the other hand, AIM and Skype are more resilient in terms of packet loss ($\geq$ 11%) compared to MSN Messenger whose tolerable loss rate is 9%. The comparative analysis in Figure 7 shows that none of the applications excels in all aspects. However, if the bandwidth is of the major concern, Skype and MSN Messenger would be the best choices; otherwise, AIM is preferred as its minimum bandwidth requirement is relatively low at 60 Kbps.

4.3.3 *Network Gaming*

In this subsection, we present an intra-genre analysis of the two RPG games [6]. The inter-genre analysis is presented in Section 4.4. Figure 6 shows that Lineage has a slightly stricter demand in terms of network delay than WoW. We confirm this difference by independent experiments in which the magnitude of the network delay is fixed and known to players. According to the players, when the RTT is large, it causes perceivable lag in the characters' movements in Lineage, but it has little impact on WoW. Given the same large RTT in the game play, Lineage players can sense the movement delay, whereas WoW players barely notice the difference if they are not interacting with other players or the environment. We attribute the cause to the fact that WoW game clients support local simulations, which allow players to control their characters without step-by-step acknowledgement from the server unless any form of interaction is performed. Moreover, WoW implements dead reckoning [25], so other characters would continue to move on the screen even if the communications with the server are cut temporarily.

By contrast, WoW requires a higher bandwidth than Lineage to support unobstructed gaming. We believe this is because WoW supports a higher degree of local simulations and dead reckoning, both of which require more game states to be transmitted by the server in real time. In addition, WoW's graphical effects are richer and more sophisticated than those of Lineage, so the states of more game objects have to be transmitted over the Internet. To name a few, the flying of weapons, such as axes being thrown, and the environmental updates, such as the debris resulting from bomb explosions.

4.4 Cross-Service Assessment

We now present a cross-service assessment of the QoS

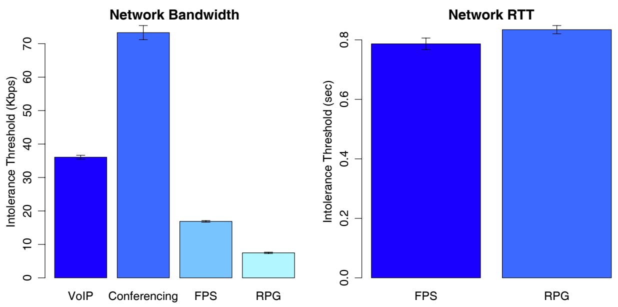

Figure 8: Comparison of intolerance thresholds of different networked multimedia services

needs of the four compared networked multimedia services[4]. Figure 8 summarizes the minimum QoS needs of each service. They are computed by averaging the needs of the applications that provide the same service. In terms of network bandwidth, the results are not surprising. Conferencing is the most demanding in terms of network bandwidth, and VoIP, FPS games, and RPG games rank 2nd, 3rd, and 4th respectively. It has been observed that FPS games require more bandwidth than RPG games because they are fast-paced, i.e., players only have sub-seconds to react at any time [7].Coincidentally, the relative bandwidth requirements of conferencing, VoIP, FPS games, and RPG games are roughly 8 : 4 : 2 : 1 (70 Kbps, 35 Kbps, 17 Kbps, and 8 Kbps) respectively. Although this may be a coincidence, it can serve as a handy guideline for network planning and resource arbitration purposes. For instance, the ratios indicate that the bandwidth required for one FPS game session can be alternatively used to serve two RPG game sessions at the minimum acceptable level of gaming experience.

To our surprise, the network delay requirements for FPS and RPG games are similar. We discovered that, although UT is a fast-paced FPS game, it supports a high degree of local simulations, which can reduce the impact of large delays unless character interactions are involved. The result indicates that local simulations can be very helpful in reducing the impact of network latency, even in fast-paced games like FPS games. Moreover, even if local simulations and dead reckoning are implemented, 0.8 seconds might be the longest tolerable round-trip time for FPS and RPG games. This is plausible because interaction between game characters always involves data exchange between game peers, such as actions performed by other players and environmental updates of the game world. We plan to conduct more studies to verify the observations regarding the games' delay requirements.

5. DISCUSSION

During the development and pilot study of our framework, we have identified the following issues worth to be discussed and investigated further.

5.1 Comparison to Mean Opinion Scores

A common, intuitive interpretation of the proposed framework is that it gives an alternative approach for assessing the quality of multimedia systems. Thus, one may wonder how the framework performs compared with the traditional MOS method. However, as we have explained in Section 1, the framework is not designed to replace MOS. Rather, we expect it works a *complementary* methodology to MOS, that is, while the MOS method is used to quantify the QoE provided by a system, the proposed framework is used to measure the system's *minimum QoS requirements* with which the QoE it provides would be satisfactory. In other words, MOS cannot be used to detect such requirements, and our framework is not designed to quantify the QoE levels of multimedia systems. This is the reason why we do not provide a comparison of the proposed framework and MOS.

5.2 Interpretation of Intolerance Threshold

The interpretation of "barely acceptable" quality may vary across different participants. While most users may be moderately aware of service quality degradation, some may be more tolerant and avoid clicking until the quality becomes worse than unacceptable. However, all the users can pass the behavior consistency tests as long as they maintain their own standards. To address this problem, a more sophisticated design that can detect such differences may be required.

5.3 Mapping to QoE

The "barely acceptable" quality defined by users may be *relative* rather than *absolute* compared to the QoE (Quality of Experience) users actually perceive. For example, users can easily become annoyed with quality degradation if an application provides a high quality experience under perfect network conditions, even if the current quality in an absolute sense is still considered good. The phenomenon is called the "expectation effect" [26]. We plan to explore this issue by mapping the derived intolerable quality onto the QoE scale [18].

5.4 Content-dependent Measurements

From Section 4.2.2, we can see that the minimum QoS needs may be significantly affected by the service's instantaneous workload and/or multimedia content offered. Therefore, the inferred QoS needs may be inconsistent if different multimedia content is being transmitted or the involving parties have different levels of interaction. In view of the purpose of this paper, the workload variation is not a serious issue because we care about the minimum QoS needs of a networked multimedia or a particular application rather than that of a particular content.

The proposed framework can be further extended to take the variability of workload and content complexity into account by treating the workload intensity as a parameter in addition to network QoS factors. By so doing, we can then quantify the minimum acceptable QoS needs for a networked multimedia service under any specific workload.

5.5 Multi-dimensional QoS

In our pilot study, we assumed that only one QoS factor is variable while other factors are kept constant; however, QoS degradation in real networks is multi-dimensional and correlated, i.e., insufficient bandwidth and high packet loss rates usually go hand in hand. For a multi-dimensional extension of the proposed framework, an intuitive approach is to focus on a dimension at a time while treating the other dimensions invariant. Assume that we have n QoS factors

[4] Here the RPG and FPS games are treated as different services.

in total and the magnitude of each factor is quantified by m levels, we have to repeat the experiments $(n-1)^m$ times in order to obtain the intolerance thresholds of a certain factor in different scenarios. In other words, the number of experiments required would grow exponentially as the number of QoS factors increases. Therefore, a more efficient approach is needed to address this problem. We plan to consider the extension of multi-dimension QoS factors in our future work.

5.6 Extension to Non-networking Factors

We note that though we focus on the minimum level of network QoS for real-time networked multimedia services in the pilot study, our framework is not limited to such scenarios. By replacing the network QoS factors with non-networking factors such as compression level, quantization degree, frame rate, screen size, audio volume, and so on, the framework can be easily extended to evaluate the minimum needs of non-networking QoS factors without any changes.

5.7 Crowdsourcing Support

Since subjective QoE experiments is costly, we believe that crowdsourcing [8, 11] will be a good strategy to conduct such experiments. Our framework supports the behavior consistency tests of participants' judgements, which is the key to crowdsourcing the experiments. However, if network factors are to be evaluated, since such experiments rely on an exact control of network quality, how to make the experiments crowdsourceable so that participants can perform experiments via their own computers at their places remains a challenge. We believe that the rich media technologies, such as Flash and Silverlight, and virtualization technologies, such as Xen and VMWare, may be keys to the crowdsourcing of network experiments.

6. CONCLUSION

In this paper, we have proposed a general, cheat-proof framework that can quantify the minimum QoS needs of real-time networked multimedia services. We have intended to make the framework general so that network and multimedia researchers and practitioners can utilize it to measure the QoS needs of their own systems; the cheat proof support of the framework is also essential since not every decision from every experiment participant is trustworthy: participants may make wrong judgements any time due to tiredness, carelessness, or even deliberate willfulness. We have shown that, with our framework, even untrained, inexperienced participants can produce consistent judgments. In addition, the derived QoS needs can serve important reference and have numerous applications in the evaluations of competitive applications, application recommendation, network planning, and resource arbitration.

7. REFERENCES

[1] F. Agboma and A. Liotta. User centric assessment of mobile contents delivery. In *Proceedings of the 6th Advances in Mobile Multimedia*, pages 121–130, Dec. 2006.

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

[3] J. M. Bland and D. G. Altman. Multiple significance tests: the Bonferroni method. *British Medical Journal*, 310:170–170, 1995.

[4] A. Bouch and M. A. Sasse. Network quality of service: What do users need? In *Proceedings of the 4th International Distributed Conference*, pages 21–23, 1999.

[5] R. Braden, D. Clark, and S. Shenkar. Integrated services in the Internet architecture: an overview. RFC 1633, 1994.

[6] Y.-C. Chang, K.-T. Chen, C.-C. Wu, C.-J. Ho, and C.-L. Lei. Online game QoE evaluation using paired comparisons. In *Proceedings of IEEE CQR 2010*, June 2010.

[7] W. chang Feng, F. Chang, W. chi Feng, and J. Walpole. A traffic characterization of popular on-line games. *IEEE/ACM Transactions on Networking*, 13(3):488–500, June 2005.

[8] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei. Quadrant of Euphoria: A crowdsourcing platform for QoE assessment. *IEEE Network*, 2010.

[9] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei. A crowdsourceable QoE evaluation framework for multimedia content. In *Proceedings of ACM Multimedia 2009*, 2009.

[10] P. de Cuetos and K. W. Ross. Adaptive rate control for streaming stored fine-grained scalable video. In *Proceedings of ACM NOSSDAV'02*, pages 3–12, 2002.

[11] J. Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):176–183, 2006.

[12] P. Hsueh, P. Melville, and V. Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35. Association for Computational Linguistics, 2009.

[13] T.-Y. Huang, K.-T. Chen, and P. Huang. Tuning the redundancy control algorithm of Skype for user satisfaction. In *Proceedings of IEEE INFOCOM 2009*, April 2009.

[14] T.-Y. Huang, P. Huang, K.-T. Chen, and P.-J. Wang. Can Skype be more satisfying? – a QoE-centric study of the FEC mechanism in the internet-scale VoIP system. *IEEE Network*, 2010.

[15] ITU-T Recommandation P. 800. Methods for subjective determination of transmission quality, 1996.

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

[17] ITU-T Recommendation G.114. General recommendations on the transmission quality for an entire international telephone connection - one-way transmission time, 2003.

[18] R. Jain. Quality of experience. *IEEE Multimedia*, 11(1):96–97, Jan.-March 2004.

[19] J. K. Kies. A psychophysical evaluation of frame rate in desktop video conferencing. In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*, pages 310–314, 1997.

[20] Y. Lee, J. Lou, J. Luo, and X. Shen. An efficient packet scheduling algorithm with deadline guarantees for input-queued switches. *IEEE/ACM Transactions on Networking*, 15(1):212–225, 2007.

[21] Y. J. Liang, N. Farber, and B. Girod. Adaptive playout scheduling and loss concealment for voice communication over IP networks. *IEEE Transactions on Multimedia*, 5:532–543, 2003.

[22] J. D. McCarthy, M. A. Sasse, and D. Miras. Sharp or smooth?: Comparing the effects of quantization vs. frame rate for streamed video. In *Proceedings of CHI 2004*, pages 535–542, Mar. 2004.

[23] D. Miras. A survey of network QoS needs of advanced internet applications. Technical report, Internet2 QoS Working Group, 2002.

[24] S. Nowak and S. Rüger. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval*, pages 557–566. ACM, 2010.

[25] L. Pantel and L. C. Wolf. On the suitability of dead reckoning schemes for games. In *Proceedings of ACM NetGames'02*, 2002.

[26] A. Parasuraman, V. A. Zeithaml, and L. L. Berry. Alternative scales for measuring service quality: a comparative assessment based on psychometric and diagnostic criteria. *Journal of Retailing*, 70(3):201–230, 1994.

[27] Z. Qiao, L. Sun, N. Heilemann, and E. Ifeachor. A new method for VoIP quality of service control use combined adaptive sender rate and priority marking. In *Proceedings of IEEE ICC'04*, pages 1473–1477, 2004.

[28] B. Sat and B. W. Wah. Playout scheduling and loss-concealments in VoIP for optimizing conversational voice communication quality. In *Proceedings of ACM Multimedia'07*, pages 137–146, 2007.

[29] C. J. Sreenan, J.-C. Chen, P. Agrawal, and B. Narendran. Delay reduction techniques for playout buffering. *IEEE Transactions on Multimedia*, 2:88–100, 2000.

[30] S. Stevens. Mathematics, measurement, and psychophysics. *Handbook of experimental psychology*, pages 1–49, 1951.

[31] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[32] C.-C. Wu, K.-T. Chen, C.-Y. Huang, and C.-L. Lei. An empirical evaluation of VoIP playout buffer dimensioning in Skype, Google Talk, and MSN Messenger. In *Proceedings of ACM NOSSDAV 2009*, 2009.