Online Game QoE Evaluation using Paired Comparisons

Yu-Chun Chang[1], Kuan-Ta Chen[2†], Chen-Chi Wu[1], Chien-Ju Ho[3], and Chin-Laung Lei[1]

[1]Department of Electrical Engineering, National Taiwan University
[2]Institute of Information Science, Academia Sinica
[3]Department of Computer Science and Information Engineering, National Taiwan University
{congo,bipa}@fractal.ee.ntu.edu.tw, ktchen@iis.sinica.edu.tw, kinkin@csie.ntu.edu.tw, lei@cc.ee.ntu.edu.tw

Abstract—To satisfy players' gaming experience, there is a strong need for a technique that can measure a game's quality systematically, efficiently, and reliably. In this paper, we propose to use paired comparisons and probabilistic choice models to quantify online games' QoE under various network situations. The advantages of our methodology over the traditional MOS ratings are 1) the rating procedure is simpler thus less burden is on experiment participants, 2) it derives ratio-scale scores, and 3) it enables systematic verification of participants' inputs.

As a demonstration, we apply our methodology to evaluate three popular FPS (first-person-shooter) games, namely, Alien Arena (Alien), Halo, and Unreal Tournament (UT), and investigate their network robustness. The results indicate that Halo performs the best in terms of their network robustness against packet delay and loss. However, if we take the degree of the games' sophistication into account, we consider that the robustness of UT against downlink delays should able be improved. We also show that our methodology can be a helpful tool for making decisions about design alternatives, such as how dead reckoning algorithms and time synchronization mechanisms should be implemented.

I. INTRODUCTION

Online gaming has been shown a profitable killer application of the Internet. To provide better gaming service to players, game developers and network engineers endeavor to improve the quality of game software and network infrastructure respectively. As their ultimate goal is to satisfy the players' gaming experience, there is a strong need for a technique that can *measure a game's quality systematically, efficiently, and reliably in a specific environment*. With such a technique, we can evaluate how good a game's quality is and estimate how much users will enjoy playing the game in certain situations by integrating a prediction model.

Since there is no exact definition of a game's quality, which may even include the game's story, art, scoring rules, and user interface; in this work, *we restrict the quality to the aspects of realtimeliness and interactivity during game play*.

The two aspects are especially critical to online games because they can easily be degraded due to network impairment (e.g., network delay and packet loss), which is one of the most uncontrollable factors affecting a game's playability. In short, *we will focus on how to efficiently and reliably measure a game's realtimeliness and interactivity in various network scenarios*. We begin with a number of examples to illustrate how such quality measurement techniques can help us pursue a better gaming experience:

1) **Game development.** If the developed game performs better or worse than competitors in terms of network performance? Also, if we have several design alternatives available for adoption, which one design will provide the best experience to gamers?

2) **Game server deployment.** How should the game servers and network links be planned in order to keep a balance between deployment cost and customer satisfaction?

3) **Game play.** if two access networks are available at a player's side, e.g., a WiFi network connecting to the Internet via ADSL (Asymmetric Digital Subscriber Link) vs. a WiMax connection, which one network will provide a more satisfactory gaming experience?

A quality measurement technique like *paired comparison*, which we will introduce shortly, is very helpful for answering the above questions. Hereafter, we shall call the game's quality we plan to measure QoE (Quality of Experience), as it indicates a user's subjective satisfaction degree in game play. This term is related to the more commonly used QoS (Quality of Service), which refers to an objective system performance metric, such as the bandwidth, network delay, and loss rate of a communication network.

To evaluate an application's QoE, the most commonly used methodology is called the MOS (Mean Opinion Score) rating test. In a MOS test, subjects are asked to give a rating between Bad (the worst) and Excellent (the best) to grade the quality of a stimulus, and the overall rating of the stimulus is obtained by averaging the scores in repeated tests. However, there are some problems on traditional MOS rating: 1) the rating procedure is somewhat difficult to experiment participants due to the scales of rating that cannot be concretely defined, 2) different participants may have dissimilar interpretations about the scales [23], and 3) it is hard to detect problematic

TABLE I
A COMPARISON OF PAIRED COMPARISON AND COMMONLY USED METHODS IN GAME QoE STUDIES

	Generalizable	Judgement difficulty	Ratio-scale scores	Input verifiable
Paired comparison	yes	low	yes	yes
MOS ratings	yes	high	no	no
Objective performance	no	N/A	no	no

inputs since we do not know whether the participants paid full attention in scoring procedures or they just give ratings perfunctorily. Paired comparison is another method considered to evaluate an application's QoE. In [10], we proposed a crowdsourceable framework based on paired comparison to quantify the QoE of multimedia content. We show that paired comparison is an effective methodology, as it does not have the problems of the MOS scoring, while keeping its advantages. In a paired-comparison test, a participant is simply asked to compare two stimuli at a time, and vote (decide) whose quality is better based on his/her perception. It can be seen that the decision is simpler than the MOS method as the five-scale rating has been reduced to a dichotomous choice. We summarize the distinct features of pair comparison and other commonly used evaluation methods in QoE studies, i.e., the MOS ratings and objective in-game performance method, in Table I.

In this paper, we propose to use paired comparisons for evaluating online games' QoE in various network scenarios. As a demonstration, we apply the methodology to evaluate three popular FPS (first-person-shooter) games, namely, Alien Arena (Alien) [1], Halo [2], and Unreal Tournament (UT) [3], and investigate their network robustness property. We shall use the Bradley-Terry-Luce (BTL) model to analyze the paired comparison results and obtain the ratio-scale magnitudes as the game's QoE scores. We defer an overview of the BTL model to Section III.

Our contribution in this work is two-fold:

1) We propose to jointly use paired comparisons and probabilistic choice models to quantify online games's QoE under various network situations. The advantages of our methodology over the traditional MOS ratings are that 1) the rating procedure is simpler thus less burden is on experiment participants, 2) it derives ratio-scale scores, and 3) it enables systematic verification of participants' inputs.

2) We apply the proposed methodology to evaluate the network robustness of three popular FPS games. The results manifest that the methodology enables us to summarize and compare the QoE of the games in different network scenarios. The analysis results show that the three games exhibit very different behavior in reaction to network impairment. In addition, according to the games' robustness against delay jitters in either directions, we can even infer how the time synchronization mechanism is implemented in each game.

The remainder of this paper is organized as follows. Section II describes related works. We present the BTL model, which is used to extract ratio-scale QoE scores from paired-comparison results, in Section III. In Section IV, we describe how we setup the network environments for evaluating the QoE of FPS games. In Section V, we discuss the effect of network impairment on the games' QoE scores and its implications. Finally, Section VI draws our conclusion.

II. RELATED WORK

A. Game QoE Studies

A number of previous works have been done to assess the QoE provided by online games in various network situations. Those studies can be categorized into *experimental* and *observational studies*. Experimental studies estimate games' QoE based on users' perception measures or their game scores in controlled environments, while observational studies infer users' satisfaction degree from real-life traces [9]. In the following we review the experimental studies as they are closely related to our work.

Henderson et al. discussed methods of soliciting player feedback in [14, 15, 19]. Armitage investigated latency tolerance of players in first-person-shooter games in [4, 5]. Furthermore, experimental studies can be divided into *subjective* and *objective studies*. Subjective studies are mostly based on MOS scores or descriptive reports about users' perceptions, e.g., [7, 22, 25]. For example, Quax et al. evaluated the influence of small amounts of delay and jitters on Unreal Tournament 2003 [22]. Objective studies are based on users' in-game performance, such as the number of kills in shooting games, the time taken to complete each lap in racing games, or the capital accumulated in strategy games. For instance, Beigbeder et al. found that typical ranges of packet loss and latency do not significantly affect the outcome of the game Unreal Tournament 2003 [7], while Sheldon et al. concluded that, overall, high latency has a negligible effect on the outcome of Warcraft III [25].

B. Studies based on Paired Comparisons

Paired comparison takes advantage of simple comparative judgements to prioritize a set of stimuli, and is able to quantify the preferences of the stimuli by adopting probabilistic choice modeling. Paired comparison is used in various domains, notably decision making and psychometric testing. Analytic Hierarchy Process (AHP) [24] is a well-known application of paired comparison. AHP uses the preference priorities extracted from paired comparison results to construct a hierarchical framework that can assist people making complex decisions. Paired comparison is also used in the ranking of universities [13], rating of celebrities [16], and various subjective sensation measurement, such as pain [18], sound quality [11], and taste of food [21].

III. PROBABILISTIC CHOICE MODELING

In the method of paired comparison [12], the basic measurement unit is the comparison of two stimuli. Assume that we have an experiment composed of t stimuli $T_1, ..., T_n$, thus there

are C_2^n stimulus pairs. We denote the number of comparisons for the pair (T_i, T_j) as n_{ij}, where $n_{ij} = n_{ji}$. The results of paired comparisons can be summarized by a matrix of choice frequencies, represented as $\{a_{ij}\}$, where a_{ij} denotes the number of choices the participant(s) preferring T_i over T_j and $a_{ij} + a_{ji} = n_{ij}$.

	T_1	T_2	T_3	T_4
T_1	–	a_{12}	a_{13}	a_{14}
T_2	a_{21}	–	a_{23}	a_{24}
T_3	a_{31}	a_{32}	–	a_{34}
T_4	a_{41}	a_{42}	a_{43}	–

TABLE II
AN EXAMPLE MATRIX OF CHOICE FREQUENCIES OF FOUR STIMULI.

By applying a probabilistic choice model to the paired comparison results, one can 1) verify whether the results are self-consistent, and 2) extract a ratio-scale score for each stimulus. One of the most widely used models for this purpose is the Bradley-Terry-Luce (BTL) model [8, 17], which predicts P_{ij}, the probability of choosing T_i over T_j, as a function associated with the true ratings of the two stimuli:

$$P_{ij} = \frac{\pi(T_i)}{\pi(T_i) + \pi(T_j)} = \frac{e^{u(T_i)-u(T_j)}}{1 + e^{u(T_i)-u(T_j)}},$$

where $\pi(T_i)$ is the estimated score of the stimulus T_i and $u(T_i) = \log \pi(T_i)$, which can be obtained by using the maximum likelihood estimation.

To verify whether the paired comparison results are consistent, one can 1) check the stochastic transitivity properties, 2) use Kendall's u-coefficient, and 3) check the goodness-of-fit of the BTL model. The stochastic transitivity method consists checks of three variants of transitivity property, including the weak (WST), moderate (MST), and strong (SST) stochastic transitivity. The three versions of transitivity imply that if $P_{ij} \geq 0.5$ and $P_{jk} \geq 0.5$, then

$$P_{ik} \geq \begin{cases} 0.5 & \text{(WST)}, \\ min\{P_{ij}, P_{jk}\} & \text{(MST)}, \\ max\{P_{ij}, P_{jk}\} & \text{(SST)}, \end{cases}$$

for all stimuli T_i, T_j, and T_k. Among the three properties, WST is the least restrictive one. Systematic violations of WST indicate that the paired comparison results cannot be integrated into a global preference ordering. Less severe violations of MST or SST can help decide whether probabilistic choice modeling is suitable for analyzing the choice frequencies.

The Kendall's u-coefficient is defined as

$$u = \frac{2 \sum_{i \neq j} \binom{a_{ij}}{2}}{\binom{m}{2}\binom{n}{2}} - 1.$$

If the subjects are in complete agreement, there will be $\binom{n}{2}$ cells containing the number m and $\binom{n}{2}$ cells being zero in the matrix of choice frequencies, and thus $u = 1$. As the number of agreements decreases, u decreases as well. The minimum agreement occurs when each cell is $m/2$ if m is even, and $(m \pm 1)/2$ if m is odd, so the minimum equals $-1/(m-1)$ if m is even, and $-1/m$ if m is odd, respectively.

The third method for consistency check is validating the goodness-of-fit of the BTL model. To do so, we compare
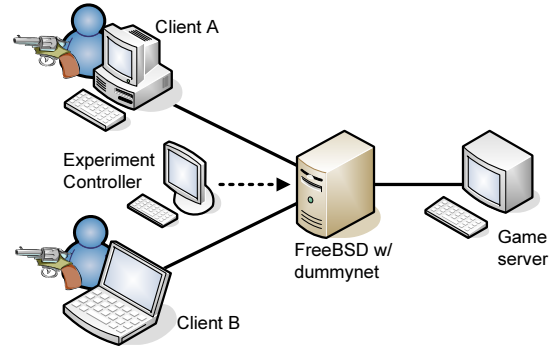

Fig. 1. The experiment setup for evaluating FPS games' QoE

the likelihood L_0 of the fitted model and the likelihood L of the unrestricted model which perfectly fits to the choice frequencies. The test statistic $-2\log(L_0/L)$ is approximately χ^2-distributed with (n-1) degrees of freedom.

IV. EXPERIMENT METHODOLOGY

In this section, we present our experiment methodology for evaluating an online game's QoE under different network settings.

In order to fully control the network conditions, we conduct our experiments in LAN. As depicted in Fig. 1, we set up two game clients and one game server in all of which three FPS games, Alien Arena 2008, Halo, and Unreal Tournament 2004, are installed. Two participants, sitting by the two game clients respectively, are asked to connect to the server and join the same game. Each game is configured in the Deathmatch mode, where players have to kill every other character they meet or their characters will be killed. We set up another FreeBSD machine as the experiment controller, which is used to configure the `dummynet` running on the gateway machine during experiment.

To facilitate paired comparisons, we need to provide the participants two sets of network conditions during their game play. Since a game can only be played with one network condition at a time, our design is to make the network condition switches between two configurations over time. Also, when participants are playing games, they may be too busy to manually trigger a network-configuration-change request. Therefore, we adopt an *automatic stimulus switching strategy*. That is, during an experiment, the network condition will be automatically and continually switched between two configurations every t seconds. We notify the participants the current stimulus by displaying a big "A" or "B" mark on the screen of the experiment controller, which resides in front of both participants. Each of the two marks denotes one network configuration (i.e., stimulus) being compared. To avoid the within-pair ordering effect, how a network setting maps to either mark "A" or "B" is randomly chosen before each test. When the participants are playing, they may sense the difference in gaming experience with different configurations. After they can conclude whether "A" or "B" configuration is more satisfactory, they can press a specific key on the keyboard of the experiment controller and enter their decisions. A test will continue until both of the participants make their choices.

	Settings		# Comparisons
Delay	0 ms, 200 ms, 400 ms, 600 ms, 800 ms, 1000 ms	uplink	600
		downlink	690
Loss	0%, 10%, 20%, 30%	uplink	288
		downlink	252
Jitter	0 ms, 250 ms, 500 ms	uplink	72
		downlink	78

In our experiments, we use $t = 5$ seconds to keep the experiment efficient (in terms of time) and allow sufficient time for the participants to perceive the game's smoothness and interactivity under both network configurations. Note that we have two participants in a test simply because a minimum of two players are required to form a deathmatch game. More participants can take an experiment at the same time as long as the game's design allows (i.e., Alien and Halo both allow a maximum of 16 players join a game simultaneously).

V. EXPERIMENT RESULTS

In this section, we present our experiment results of the FPS games' QoE under different network conditions. First we summarize the paired comparison results collected from our experiments. After the results' consistency is verified, we then investigate the effect of network delay, loss, and delay jitter on the QoE of the FPS games.

A. Data Summary

We carry out three sets of experiments. In each of the experiments, we change network delay, loss rate, and delay jitter respectively and set the other factors to their respective ideal settings, i.e., no delay, zero loss rate, and no delay variations. As network impairment may have different impacts when it occurs in different links, each set of experiments was repeated twice with the impairment applied to the uplink and downlink respectively, where the uplink indicates the network path from the game client to the server, and the downlink indicates the path in the opposite direction. We asked a total of five college students to take the experiments in different time periods, where each test exactly contains two participants.

A summary of the experiment settings and the number of tests performed are listed in Table III. The numbers of settings for the delay, loss, and jitter experiments are 6, 4, and 3, respectively. These numbers are chosen because we are more interested in the effect of delay and more settings allow us to inspect the QoE behavior in more depth. However, more settings indicate that more tests (i.e., comparisons) are needed in order to achieve a preference rating with high confidence (i.e., a narrow confidence band). This is also the reason why the number of comparisons in the delay experiments is much higher than that in other experiments.

Before applying the BTL model to analyze the paired comparison results, we perform the consistency checks for our data in order to make sure the participants did not make decisions arbitrarily. The consistency analysis results are presented in Table IV. We can see that the numbers of WST, MST, and SST violations are very small compared with the number of comparisons. In addition, if we consider both the Kendall's

Game	Factor	Link	WST	MST	SST	Kendall	p-BTL
Alien	delay	uplink	0	2	7	0.35	0.33
		downlink	0	0	3	0.62	0.80
	loss	uplink	0	0	1	0.40	0.12
		downlink	0	0	0	0.74	0.86
	jitter	uplink	0	1	1	-0.04	0.26
		downlink	0	0	0	1.00	0.02
Halo	delay	uplink	0	0	6	0.36	0.35
		downlink	2	2	5	0.53	0.15
	loss	uplink	0	0	0	0.25	0.98
		downlink	0	1	3	0.22	0.03
	jitter	uplink	0	0	0	0.55	0.44
		downlink	0	1	1	0.32	0.05
UT	delay	uplink	0	1	10	0.42	0.01
		downlink	0	2	5	0.56	0.11
	loss	uplink	0	1	1	0.28	0.01
		downlink	0	0	2	0.43	0.17
	jitter	uplink	0	0	0	1.00	0.02
		downlink	0	0	0	-0.08	1.00

u-coefficient and the p-value of the goodness-of-fit test for the BTL model, almost every experiment pass the tests except for some of the delay jitter experiments. The failure in passing the consistency check in delay jitter experiments is due to the participants cannot figure out the difference due to different jitter settings, which we will explain in Section V-D.

B. Effect of Network Delay

The games' QoE scores with respect to different network delays are shown in Fig. 2 in which dot lines stand for 95% confidence bands of QoE scores. Since the QoE scores are on a ratio scale, it can be proportionally scaled without losing the comparative information. To make the plot easier to read, we normalize the QoE scores on each plot by making the highest QoE score 1. Therefore, all the QoE scores are within the range of 0 to 1. Since we include the ideal network condition in each experiment, without loss of generality, we can thus assume the best QoE a game can provide is 1, and inspect how the QoE score degrades due to worse network conditions. Please note that our methodology does not allow us to compare the absolute QoE scores across games as we only have a game's relative quality in different network scenarios. Instead, the QoE scores we obtained enable us to observe the "network robustness" property of a game. That is, *how resilient is a game's QoE against network impairment*. The network robustness property can be observed from the relationship between a game's QoE score and the corresponding network setting.

Back to Fig. 2, overall, we can see that the effect of delay is different on the uplink and downlink. From the trend the uplink delays have a relatively less severe effect on the QoE, while the downlink delays can easily result in an unacceptable QoE for all the games. We believe that this discrepancy is due to the different nature of the data conveyed by uplink and downlink packets.

Uplink packets primarily contain a player's inputs to the server; thus, a longer delay would consistently lead to a longer command response time and thus a lower interactivity. Since game clients can provide immediate feedback of a player's inputs on his/her screen, the impact of uplink delay can be somewhat eliminated. However, for the player's actions that make changes to the environment or other characters, we still need to wait for the corresponding responses from the
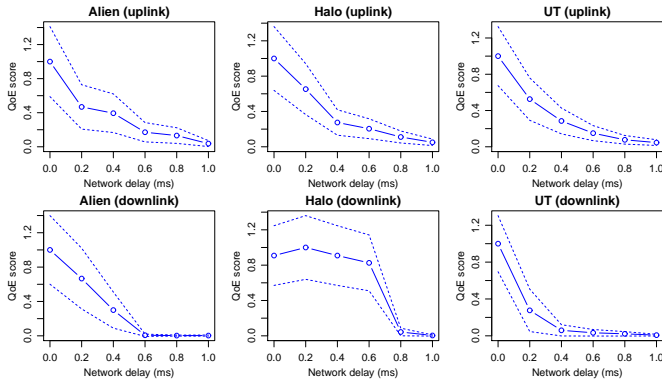
Fig. 2. Games' QoE scores vs. network delay

Fig. 3. Games' QoE scores vs. network loss rate

game server. Therefore, the uplink delay still has significant degradation effect on QoE.

On the other hand, downlink packets primarily contain data in two categories: 1) the environmental changes made by other players, and 2) the actions made by other players. If downlink packets cannot arrive at the game client continuously and smoothly, the client's screen will be freezed as no new state updates of the game world are received. There have been some proposals to remedy such situations, e.g., dead reckoning [20]; however, the effect of such solutions is limited, as future actions of other players are highly unpredictable. We can see that the QoE scores of the three games are degraded to nearly zero when the downlink delays are no shorter than 0.6 sec (Alien), 0.8 sec (Halo), and 0.4 sec (UT), respectively. Note that the QoE score of Halo remains statistically unchanged when downlink delays are no longer than 0.6 sec. We believe that this phenomenon implies that Halo implements certain kind of dead reckoning technique so that even moderate downlink delays are not aware by the gamers. In contrast, UT is most sensitive to downlink delay increase, which indicates that it does not implement appropriate local prediction techniques to cope with high network delays.

If we compare the three games, we can see that the impact of uplink delays on the games are basically the same. This should be due to that all the games have provided immediate feedback of the player's actions, as this might be the best remedy game designers can do. Meanwhile, various prediction techniques exist for mitigating the impact of downlink delays. Based on our results, we believe that Halo does the best in overcoming large downlink delays, while UT performs the worst from this perspective.

C. Effect of Network Loss

We show the games' QoE scores with respect to different network loss rates in Fig. 3. Similar to the effect of network delays, we find that Halo performs the best no matter uplink or downlink network loss is injected. For Alien and UT, a loss rate equal to or higher than 10% in either direction can easily make the game unplayable, except the uplink loss case for UT, which seems to tolerate an uplink loss rate up to 20%. On the contrary, Halo exhibits an excellent capability in coping with the impact of packet loss, especially for downlink loss, where our subjects cannot even systematically distinguish the conditions of no loss and with 20% loss rate.
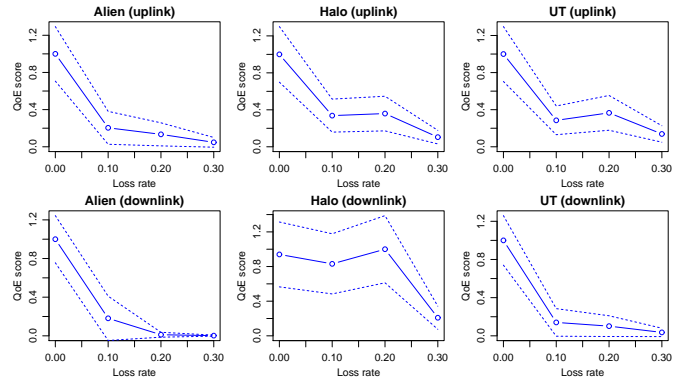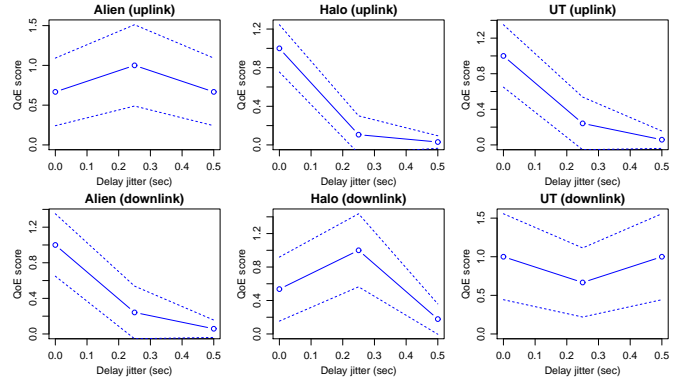

Fig. 4. Games' QoE scores vs. network delay jitter

D. Effect of Delay Jitter

According to our earlier studies [9], delay jitter (variability) also has a significant impact on game users' satisfaction. Therefore, we also study the effect of delay jitters. We plot the relationship between the games' QoE scores and delay jitters in Fig. 4. Interestingly, we can see that each of the games is robust to delay jitters in one direction, and susceptible to delay jitters in another direction. Specifically, Alien is very sensitive to downlink delay jitters but robust to uplink jitters. In contrast, Halo and UT are sensitive to uplink delay jitters but insensible to downlink jitters. We believe that this behavior is *due to the different locations where a game implements its time synchronization mechanism between the clients and the server* [6]. The time synchronization mechanism is necessary in that it 1) keeps the game states in each client consistent, 2) maintains fairness so that a faster client will not gain any benefit, and 3) prevents time-based cheating attacks. With this mechanism, when expected packets sent from other peers arrive late, the peer which is responsible for synchronization will hold the current game state for some period, i.e., introduce a "local lag," in order to maintain a consistent game view across participating peers. From our results, only the downlink delay jitter affects Alien's QoE significantly, which indicates that Alien implements the time synchronization mechanism in its game client. On the other hand, Halo and UT should have done their time synchronization work on the game server, thus only uplink delay jitter impacts the games' playability.

From Fig. 4 we also obtain that on which peer the time synchronization should be implemented seems merely a tradeoff between the robustness against uplink jitters and that against

downlink jitters. However, considering uplink bandwidth is usually more restricted than downlink bandwidth in users' access networks, the robustness against uplink delay variability seems more important than that against downlink jitters. From this perspective, Alien's design is better than the other two games. Although the design of game architecture is not our goal in this work, here we demonstrate that our methodology can be a helpful tool for decision making between design alternatives.

VI. CONCLUSION AND FUTURE WORK

From our experiment results in Section V, it may be mistaken that Halo is better than Alien and UT in terms of their network design and performance. In fact, such conclusions are difficult, if not impossible, to make because the requirement for network support of different games can be very different due to their variety in game design, scene complexity, game pace, game rules, playing strategy, and so on.

More concretely, according to our experiment participants, Halo's game pace is significantly slower than Alien and UT. Also, the scene complexity and special effects in Alien is far more sophisticated than those in the other two. The special effects such as weapon firing and bullet flying are impressive in Alien. UT also has splendid special effects, while Halo only provides relatively primitive effects. Therefore, it is reasonable that Halo exhibits the best robustness to network impairments since it scene complexity is the lowest and therefore it should have the least requirement for network data delivery.

If we take the degree of the games' sophistication into account, we consider that the robustness of UT against downlink delays should able be improved compared with that of Alien and Halo (cf. Fig. 2). Although currently our methodology does not provide a numeric metric for a game's network performance, we have shown in Section V-D that it can be a helpful tool for making decisions regarding network design and functionalities, such as how a dead reckoning algorithm should be designed and where the time synchronization mechanism should be implemented.

In the future, we are to continue our studies on the network robustness of online games. First, we aim to conduct more experiments and summarize the similarity of and difference between games of the same genre. Next, we will expand our study scope and include the comparison of different game genres into consideration. We will target a goal to understand the general requirement for network support of different game genres, and in consequence derive a network requirement profile for each of the game genres.

REFERENCES

[1] "Alien Arena." [Online]. Available: http://icculus.org/alienarena/rpa/aquire.html

[2] "Halo Combat Evolved." [Online]. Available: http://halo.wikia.com/wiki/Halo:_Combat_Evolved

[3] "Unreal Tournament 2004." [Online]. Available: http://www.unrealtournament2003.com/ut2004/

[4] G. Armitage, "An experimental estimation of latency sensitivity in multiplayer quake 3," in *The 11th IEEE International Conference on Networks*, 2003, pp. 137–141.

[5] G. Armitage and L. Stewart, "Limitations of using real-world, public servers to estimate jitter tolerance of first person shooter games," in *Proceedings of ACM SIGCHI ACE 2004 Conference*, 2004, pp. 257–262.

[6] N. E. Baughman and B. N. Levine, "Cheat-proof playout for centralized and distributed online games," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, AK, Apr. 2001.

[7] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool, "The effects of loss and latency on user performance in Unreal Tournament 2003," in *Proceedings of NetGames'04*. ACM Press, 2004, pp. 144–151.

[8] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[9] K.-T. Chen, P. Huang, and C.-L. Lei, "Effect of network quality on player departure behavior in online games," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 5, pp. 593–606, May 2009.

[10] 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.

[11] S. Choisel and F. Wickelmaier, "Evaluation of multichannel reproduced sound: Scaling auditory attributes underlying listener preference," *The Journal of the Acoustical Society of America*, vol. 121, no. 1, pp. 388–400, 2007.

[12] H. A. David, *The Method of Paired Comparisons*. Oxford University Press, 1988.

[13] R. Dittrich, R. Hatzinger, and W. Katzenbeisser, "Modelling the effect of subject-specific covariates in paired comparison studies with an application to university rankings," *Journal of the Royal Statistical Society (Series C): Applied Statistics*, vol. 47, no. 4, pp. 511–525, 1998.

[14] T. Henderson, "Latency and user behaviour on a multiplayer game server," in *Proceedings of the Third International COST264 Workshop on Networked Group Communication*. Springer-Verlag, 2001, pp. 1–13.

[15] T. Henderson and S. Bhatti, "Modelling user behaviour in networked games," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001, pp. 212–220.

[16] C. L. Knott and M. S. James, "An alternate approach to developing a total celebrity endorser rating model using the analytic hierarchy process," *International Transactions in Operational Research*, vol. 11, no. 1, pp. 87–95, 2004.

[17] R. D. Luce, *Individual Choice Behavior: A Theoretical Analysis*. New York: Wiley, 1959.

[18] J. N. S. Matthews and K. P. Morris, "An application of bradley-terry-type models to the measurement of pain," *Applied Statistics*, vol. 44, pp. 243–255, 1995.

[19] M. Oliveira and T. Henderson, "What online gamers really think of the internet?" in *Proceedings of the 2nd workshop on Network and system support for games*. ACM, 2003, pp. 185–193.

[20] L. Pantel and L. Wolf, "On the suitability of dead reckoning schemes for games," in *Proceedings of NetGames'09*, 2002, pp. 79–84.

[21] N. L. Powers and R. M. Pangborn, "Paired comparison and time-intensity measurements of the sensory properties of beverages and gelatins containing sucrose or synthetic sweeteners," *Journal of Food Science*, vol. 43, no. 1, pp. 41–46, 1978.

[22] P. Quax, P. Monsieurs, W. Lamotte, D. D. Vleeschauwer, and N. Degrande, "Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game," in *Proceedings of ACM SIGCOMM 2004 workshops on NetGames '04*. ACM Press, 2004, pp. 152–156.

[23] P. Rossi, Z. Gilula, and G. Allenby, "Overcoming Scale Usage Heterogeneity: A Bayesian Hierarchical Approach," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 20–31, 2001.

[24] T. L. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234–281, 1977.

[25] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu, "The effect of latency on user performance in Warcraft III," in *Proceedings of NetGames'03*. ACM Press, 2003, pp. 3–14.