Cloud Mobile Gaming: Modeling and Measuring User Experience in Mobile Wireless Networks

Shaoxuan Wang
shaoxuan@ece.ucsd.edu

Sujit Dey
dey@ece.ucsd.edu

Mobile Systems Design Lab
Department of Electrical and Computer Engineering
University of California, San Diego

With the evolution of mobile devices and networks, and the growing trend of mobile Internet access, rich multi-player gaming using mobile devices, similar to PC-based Internet games, has tremendous potential and interest. However, the current client-server architecture for PC-based Internet games, where most of the storage and computational burden of the game lies with the client device, does not work with mobile devices, constraining mobile gaming to either downloadable, single player games, or very light non-interactive versions of the rich multi-player Internet games. In this article, we study a cloud server based mobile gaming approach, termed Cloud Mobile Gaming (CMG), where the burden of executing the gaming engine is put on cloud servers, and the mobile devices just communicate the users' gaming commands to the servers. We analyze the factors affecting the Quality of user Experience (QoE) using the CMG approach, including the game genres, video settings, the conditions of server and client, and the conditions of the wireless network. A Mobile Gaming User Experience (MGUE) model is developed and validated through controlled subjective testing. We also develop a prototype for in-service measuring MGUE, and then use it to characterize user experience achievable using the CMG approach in a mobile cellular network. The MGUE model developed and validated in this article will be helpful for researchers and service providers to develop and assess the performance of future cloud mobile gaming techniques and services.

I. Introduction

The emergence of new and more capable mobile devices, including smart phones, tablets, and netbooks, together with the steady deployment of broadband wireless networks, is making mobile access to rich Internet sites a reality. The above progress opens up a new possibility: ability to play rich Internet games, produced for PCs on wireline networks, from mobile devices. Enabling mobile Internet gaming will significantly change the experience of mobile users from thin, single player gaming possible today to rich, multi-player Internet gaming experience, of their familiar games from anywhere. It will also open the possibility of mobile service providers and Internet game developers translating the tremendous growth experienced in recent years in Internet PC games to the fast emerging mobile eco-system. However, due to the inherent hardware constraint of mobile devices (memory and graphics processing), the above goal will be difficult to achieve using the current client-server gaming architecture for PC-based Internet games, where most of the storage and computational burden of the game lies with the client device.

Instead, it may be promising to investigate an alternative approach that has been gaining attention recently, where a gaming server is responsible for executing the appropriate gaming engine, and streaming the resulting gaming video to the client device. This approach is originally being explored for Internet PC-based gaming [19][20][21], as it enables PC users to play any games on-demand from any PCs. However, it is much more promising to employ this approach with cloud computing techniques, termed Cloud Mobile Gaming (CMG) [6][7], to enable rich multiplayer Internet games on mobile devices, as it will eliminate the need for mobile devices to download and execute the memory and computation intensive gaming engines.

As shown in figure 1, Cloud Mobile Gaming server extends the traditional game content server with two additional components: game engine server and game streaming server. The mobile user issues a game command on the mobile device, which is delivered to the cloud gaming server using the wireless uplink channel, and accepted by the game engine server. The game engine server then synchronizes the gaming data and messages from the content server, and processes the gaming logic to render the raw game video. The generated raw game video is encoded by the game streaming server, and finally sent to the mobile client via the downlink

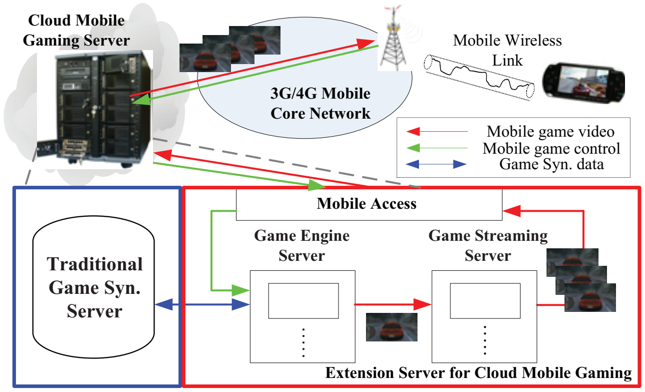

Fig. 1. Overview of Cloud Mobile Gaming Architecture and Data/Control Flow.

wireless connection.

Though the CMG approach addresses the hardware constraint of mobile device to play rich multi-player gaming, it has to cope with the challenges of mobile wireless network, considering the user control data has to be transmitted uplink from the mobile devices, and streaming video data transmitted downlink from the server, all through the bandwidth constrained and error-prone wireless links. Thereby the fundamental question arises: how good will the user experience be using the CMG approach? To answer this question, in this article, we propose a Mobile Gaming User Experience (MGUE) model to quantitatively measure the user experience of mobile gaming using the CMG approach. We develop and validate the MGUE model using controlled subjective testing, and then characterize MGUE in a commercially deployed mobile cellular network. The MGUE model developed in this article is helpful for researchers or mobile gaming service providers to assess the performance of this new CMG technique. In addition, there is also a trend to deploy many interactive multimedia applications into the cloud server, such as virtual reality and augmented reality. We believe our approach to develop MGUE model outlined in this article can be potentially used to develop Quality of Experience (QoE) models for these cloud server based applications.

The remainder of the paper is organized as follows. To derive MGUE model, section II analyzes the impairment factors that affect MGUE, and introduces Game Mean Opinion Score (GMOS) as a way to model and measure the user experience. Section III derives the impairment function of each individual impairment factor to complete the MGUE model. In section IV, we first introduce a software MGUE prototype for measuring impairment factors and calculating the corresponding GMOS score during live CMG sessions. Then we validate the MGUE model by subjective tests and enhance the MGUE with considering the cross-effects between different impairment factors. In section V, we characterize and assess the feasibility of the CMG approach in a commercial mobile cellular network, using our MGUE model. Section VI summarizes our findings and points out the future work.

II. Mobile Game User Experience (MGUE) Model

The ability to model and evaluate the QoE of a network service is important to network operators and service providers, so that they can provision for the appropriate QoE levels, monitor the QoE achieved, and take steps to improve the service as needed. QoE standardization has been actively pursued by International Telecommunications Union (ITU). Its existing standards cover the video quality metrics and tools (ITU-T J.144) [16], and quality assessment approaches for multimedia services like IPTV (ITU-T SG12) [17], VoIP (ITU-T G.107) [15], and Videophone (ITU-T G.1070) [18]. Unfortunately, there is no ITU standard quality measurement tool available for gaming. Moreover, many other video quality metrics and measurement techniques [1][2][3][4][5] cannot be directly applied to measure the user experience of (mobile) gaming, which is a highly interactive application, and where the round-trip response time has a significant impact on user experience, as opposed to the one-way nature of video. Though ITU standards of VoIP and Videophone have considered the impacts of network delay, they do not take into account round-trip delay, but only one-way delay. There has also been some work to analyze factors affecting user experience in gaming [8][9][10][11][12][13]. However, they focus on conventional PC games and do not apply to the Cloud server based Mobile Gaming (CMG) approach, which is the subject of this article.

Among existing QoE measurement methodologies, parametric model is the most commonly used way of measuring network multimedia QoE. It generally has three key components: model inputs, model assumptions, and quality estimation function. The inputs of parametric model are a group of impairment factors which affect user experience, while the model assumptions restrict the model working conditions. The quality estimation function is the essential part of the parametric model presenting the relationship between the impairment factors and model output of the predicted user experience. To develop such a

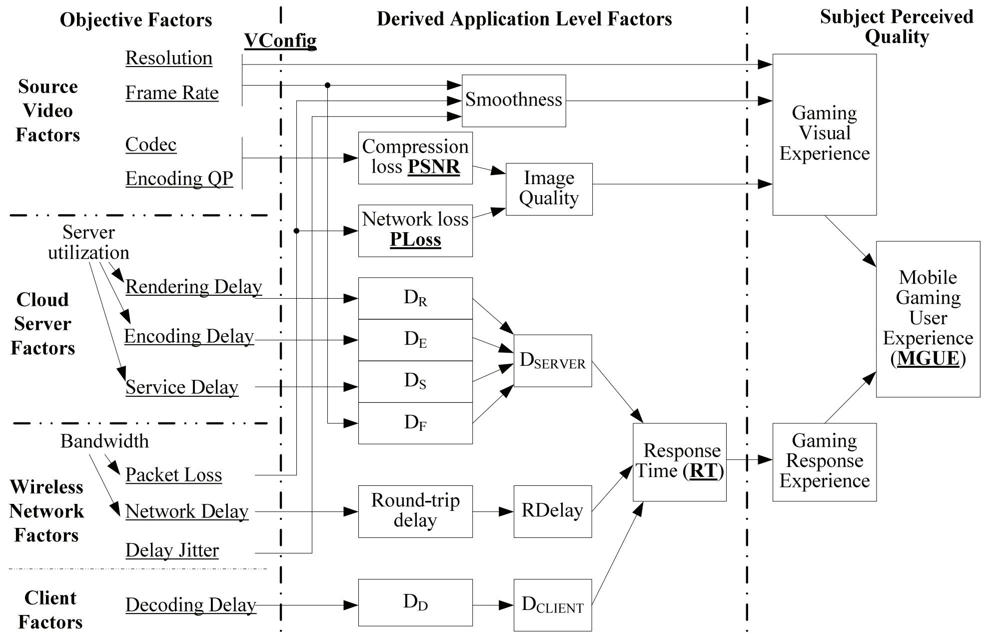

Fig. 2. Factors affecting Mobile Gaming User Experience

parametric model that can quantitatively measure MGUE, we start this section by analyzing the various factors affecting MGUE. Through giving some model assumptions and discussions, we then identify several important impairment factors as model inputs, which are sub-sequentially used to formulate quality estimation function of MGUE model. The quality estimation function of MGUE model will be finally derived through subjective experiments introduced in the next section.

II.A. Impairment Factors Affecting MGUE

User perceived MGUE would mainly depend on two subjective factors: gaming visual experience and gaming response experience. The gaming visual experience depends on the resolution, smoothness, and image quality of gaming video received by mobile client, while the gaming response experience refers to the total delay from the user control command occurring to the corresponding video frame displaying on the mobile device. User perceived video quality needs to be measured from the off-line comparison of the video displayed in the client and the reference video in the server. The exact response time can be measured from the video obtained from recording the entire visual progress when the user played the game. Both measurements of these two subjective factors are time consuming and costly. To ensure feasibility, we decided to formulate MGUE model based on objective factors that can be in-service measured.

As shown in figure 2, MGUE is affected by a number of objective factors, which can be categorized into four groups: source video factors, cloud server factors, wireless network factors, and client factors. Each of these objective factors affects the gaming visual experience and gaming response experience in a complex manner. For example, quality degradation of user received gaming video may occur during the compression process from the source, depending on compression codec and compression Quantization Parameter (QP). Besides compression loss, network packet loss due to network congestion or wireless RF conditions could also reduce the video quality. The smoothness of the video is decided by the video frame rate, network packet loss, and network delay jitter. On the other hand, the gaming response experience will be affected by various delays created in cloud server, wireless network, and mobile client.

Among all the objective factors, server utilization and network bandwidth will only affect the MGUE when they are over-utilized. Moreover, the negative effects of these two factors could be represented by other factors. For example, server over-utilization will cause unexpected increase on rendering delay, encoding delay and service delay, while wireless channel over-utilization will cause network packet loss and unexpected delay. Therefore, we do not need to consider server utilization and network bandwidth into the MGUE model. Besides the factors shown in figure 2, game genre is also an important factor that determines the contributions of all the other factors in determining the MGUE. For instance, in some games (e.g. racing games) fast response time is more crucial in determining the MGUE, while in some other games (e.g. Massively

Multiplayer Online Role-Playing Game (MMORPG)) being able to clearly see the objects, and hence sufficient video quality, is more crucial in determining the MGUE. Therefore, parametric MGUE model can be formulated as:

$$\begin{aligned} MGUE = F(&Game, Resolution, FrameRate, \\ &Codec, EncodingQP, RenderingDelay, \\ &EncodingDelay, ServiceDelay, PacketLoss, \\ &NetworkDelay, DelayJitter, DecodingDelay) \end{aligned} \quad (1).$$

However it is hard to integrate all the objective factors in equation (1) as the model inputs to determine the MGUE. Thus we present the following analysis and model assumptions which allow us to reduce the complexity of the model by reducing the number of factors.

First, video resolution and frame rate are video configurations, which are given when the video streaming starts. We group these two factors together termed as VConfig. Giving the VConfig, the encoding QP determines the video quality, but is conditional on the codec and video content. Therefore, it is hard to study and determine its impact in determining the MGUE. Instead, we use Peak Signal to Noise Ratio (PSNR), the most commonly used distortion metric, to measure the quality of compressed video at the gaming video source.

Second, as discussed earlier, user perceived gaming response experience is hard to measure. Therefore, we use a derived objective factor gaming Response Time (RT) to indicate the gaming response experience. To study and understand RT, we consider the round trip data flow in a CMG session shown in figure 3. When a user command occurs on the mobile device, it will be sent to the gaming server with a network uplink delay D_{UPLINK} (T1-T2). This command may be held in the processing queue if there are plenty of commands from other clients waiting to be processed by the cloud server. Only the first command in the queue will be processed by the server, while the other commands have to wait for a period of time called service delay D_S. Then the game engine will process this command and then generate raw game video. This will take a period of time called rendering delay D_R. However, the raw game video is not always generated just after the command is processed. There is an interval T_{FRAME} between consequential video frames, which is equal to the reciprocal of Frame Rate (FR). Due to this T_{FRAME}, there might be a delay D_F with a range from 0 to T_{FRAME} depending on interval between the time when the server processes command and the time when the next raw game video is generated. Since command is processed randomly, D_F is uniformly distributed between 0 to T_{FRAME}. Therefore the average value of D_F is $1/(2\times FR)$. Once raw game video is generated, it will be encoded and packetized in a period of time called encoding delay D_E (T3-T4). The video packets will be received after a network downlink delay $D_{DOWLINK}$ (T4-T5), and displayed onto the mobile device after a client buffering and decoding delay D_C (T5-T6). Let RDelay denotes the round-trip network delay including D_{UPLINK} and $D_{DOWNLINK}$, RT can be formulated to:

$$\begin{aligned} &RT = RDelay + D_S + D_R + D_F + D_E + D_C \\ &\text{where } D_F = 1/(2\times FR) \end{aligned} \quad (2).$$

Some of delays in equation (2) can be directly measured by CMG server and mobile client, like D_R, D_F, D_E, and D_C, while for some other delays like RDelay and D_S, we will measure them by a network probing mechanism (introduced in section IV.A).

Next, we analyze another factor, delay jitter, and explain why we will not consider it in modeling MGUE. Delay jitter denotes the network delay variation. Given an average network delay, there are two kinds of delay jitter, negative jitter and positive jitter. Negative jitter is caused by late arriving

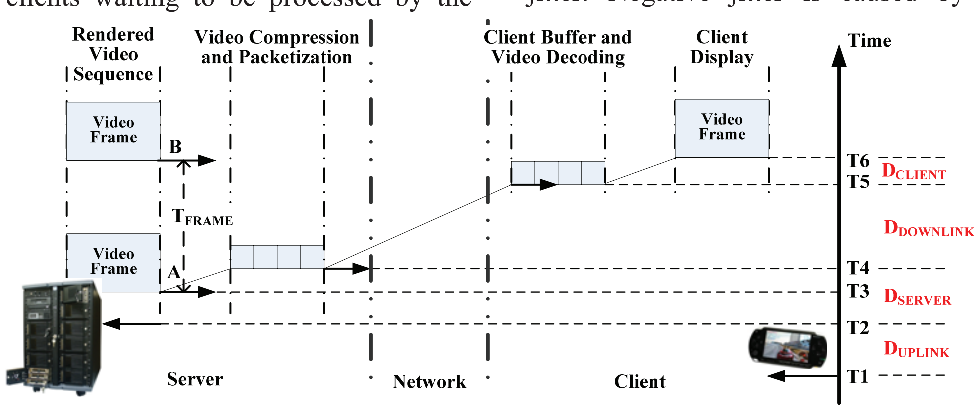

Fig. 3. Round-trip flow of gaming response time.

packets, and early arriving packets lead to positive jitter. The client buffer can eliminate positive jitter by caching the early arriving packets, while the effect of negative jitter can be represented by the larger network delay, captured in RDelay. Hence, we do not consider explicitly the effects of delay jitter in modeling MGUE.

Based on the above analysis and assumptions, we could reduce all the objective factors to five impairment factors: Game genre played, VConfig used, source video PSNR, network packet loss (PLoss), and gaming response time (RT). Thus MGUE can be formulated as:

$$MGUE = F(Game, VConfig, RT, PSNR, PLoss) \quad (3).$$

II.B. Quantitative Measurement of MGUE

Having decided model inputs (impairment factors in equation (3)), next we need define a quantitative measurement metric for MGUE. In audio and video services, the most widely used subjective quality assessment methodology is opinion rating, which is defined in ITU-T Recommendation P.800 [14]. In the subjective assessment tests, subjects are instructed to rate their perceived quality of the services according to the following opinion scales: 5(excellent), 4(good), 3(fair), 2(poor), and 1(bad). Subsequently, the arithmetic Mean of all the collected Opinion Scores, MOS [14], is used as the measure of QoE. Similarly, we introduce a Game Mean Opinion Score (GMOS) as the measurement metric for MGUE, and later in the paper, go on to develop a parametric MGUE model to quantitatively measure GMOS.

$$GMOS = F(Game, VConfig, RT, PSNR, PLoss) \quad (4).$$

Since GMOS in equation (4) is determined by 5 factors and its formulation can be a complex function, we attempt to derive simple individual functions of each factor, similar to the framework of ITU-T E-model [15] for transmission planning. Although this E-model was originally proposed for the audio transmission planning, the framework of transmission rating factor R is helpful for any transmission planning because it makes the quality judgments for good or better and poor or worse in a good statistical mapping, hence can be applied in our study. The function of MOS formulated by R can be found in [15]. We duplicate that function for our GMOS formulation:

$$GMOS = 1 + 0.035R + 7\times10^{-6} R(R-60)(100-R) \quad (5).$$

The R-factor ranges from 0 to 100 and is related to GMOS through a non-linear mapping. We first derive MGUE model with only considering the individual effect of each impairment factor, while the cross-effects of different impairment factors will be considered and added into MGUE model later in section IV. The R-factor only considering the individual effect of each impairment factor can be formulated as:

$$R = \begin{cases} 100 - \sum_i I_i & (\text{if } \sum_i I_i < 100) \\ 0 \ \ (\text{if } \sum_i I_i > 100) \end{cases} \quad (6).$$

I_i is the impairment function for each impairment factor, which indicates the individual impairment on MGUE of each impairment factor. As discussed earlier, Game genre will determine the contributions of all the other factors in determining the MGUE. Therefore, Game genre will be a parameter in every impairment function for each of the other four impairment factors (VConfig, RT, PSNR, and PLoss). Based on above discussions, we formulate the R-factor with individual impairment functions:

$$\begin{aligned} R = 100 &- I_C(Game, VConfig) - I_R(Game, RT) \\ &- I_P(Game, PSNR) - I_L(Game, PLoss) \end{aligned} \quad (7).$$

I_C includes the effect of the initial streaming video VConfig (resolution, frame rate); I_R indicates the impairment caused by Response Time; I_P represents the impairment caused by source streaming video quality PSNR; I_L covers the impairment caused by Packet Loss. The quality estimation equations (5) and (7) indicate that GMOS can be evaluated by the Game played, initial VConfig and measureable factors: RT, PSNR, and PLoss. In this section, we have decided the MGUE model inputs based on some model assumptions, and introduced a metric GMOS as a measure for MGUE. Then we have derived the quality estimation equations of MGUE model and introduced its impairment functions. In the next section, we will derive the impairment functions in quality estimation equation (7) to complete MGUE model.

III. Deriving Impairment Functions

In this section, we describe the approach we use to derive the impairment functions, I_C, I_R, I_P, and I_L. As a first step, we set up a controlled test environment, where each of the factors, VConfig, RT, PSNR, and PLoss, can be varied independently without affecting the settings of the other factors. Next, we conduct a series of MGUE subjective tests using a study group, where each test constitutes playing a game under a particular setting of one of the factors. Each study group participant provides an assessment of his/her gaming experience for each subjective test using a GMOS score. Base on our experiment results and regression analysis, we can derive the impairment function for each impairment factor. We start by describing the subjective testing process. Next, we describe how we derive the impairment functions to complete the MGUE model.

III.A. Subjective Quality Assessment Experiments

To study the effect of each of the factors on MGUE, we conduct a group of subjective quality assessment experiments. Figure 4 shows our experiment test bed. We connect the mobile device, which will be used by the study group participants, to a CMG server, directly via a network emulator, which we can use to control the network RDelay and PLoss. The user perceived RT is measured by a network probing mechanism introduced later in section IV.A, and we vary the RT by changing the RDelay via network emulator. The VConfig is varied by changing the resolution and frame rate settings in the video encoder. Similarly, the PSNR of source video is varied by appropriately changing the compression QP of the video encoder used by the CMG server. Table I shows the parameters of different factors that have been used in the subjective quality assessment experiments.

The study group was comprised of 25 students and staff at UCSD, who have prior experience of playing

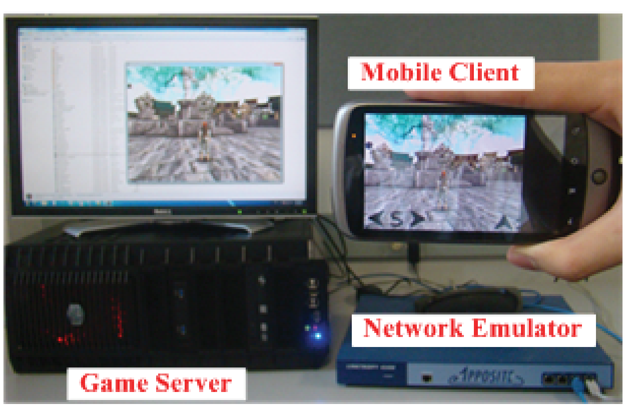

Fig. 4. Test bed of subjective quality assessment experiments

TABLE I. Experimental Parameters of Subjective Quality Assessment Experiments

Game (Type)		WoW(MMORPG), NFS(Racing), PES(Sports)
VConfig	Resolution	VGA, QVGA
	Frame Rate	[25:2:17], [15:1:5]
RDelay (ms)		[0:40:800]
PSNR (dB)		[26: 0.5: 38]
PLoss (%)		0, 0.5, 1, 2, 3, 4, 6, 8

TABLE II. GMOS Ratings and "R" Values

GMOS	R	Description
4.5—5.0	100	Excellent game, no impairment at all
4.0—4.5	80-100	Minor impairment, will not quit game
3.0—4.0	60-80	Impairment noticeable, might quit the game
2.0—3.0	40-60	Clearly impairment, usually quit the game
1.0—2.0	0-40	Annoying environment, definitely quit.

the selected games. Each study group participant played each of the three games under a certain test condition using the experimental parameters in Table I, and provided assessment of their MGUE using a GMOS score rating system shown in Table II. Finally, the results of the study group were tabulated for further analysis and derivation of the impairment functions.

III.B. Deriving Impairment Function I_C

To determine I_C, we consider the results of the subjective tests where only the Game and VConfig are changed, keeping all the other three factors at their best values, such that there is no impairment caused by them. For a given Game type, and a VConfig, we get the average GMOS score of all the participants, and use it to get the value of R from (5), and subsequently the value of I_C using (7), where the other impairment functions I_R, I_P and I_L are all 0 (as they do not cause any impairment). Table III shows the values of I_C for each of the VConfig used for each Game type. When adjacent two or more frame rates have close subjective GMOS score, we group them together (as their impacts on user experience are very close). For instance, for all three games, game users can hardly feel differences if we vary the frame rate from 25 to 16. Therefore there is only one average value of I_C of frame rate 25 to 16 for each game in different resolutions. It should also be noted that we did not study the MGUE where frame rate is below 5. One reason is that if frame rate is 5, user experience GMOS will drop below 3.0, where users cannot accept the gaming quality. Therefore, it is less necessary to study the user experience of frame

TABLE III. Value of I_C in VGA Resolution

Game		25-16	15-13	12-11	10-9	8-7	6-5
WoW	VGA	3	3	8	14	25	41
	QVGA	10	10	16	23	35	54
NFS	VGA	0	0	7	15	26	51
	QVGA	3	3	10	18	28	53
PES	VGA	0	3	10	28	46	72
	QVGA	5	7	17	32	51	70

rate below 5. Another important reason of not studying frame rate below 5 is related to the delay D_F in Response Time (RT). As discussed earlier, frame rate also affects the factor RT by D_F. When the frame rate is high, D_F is low, and thus cannot be felt by user. However when frame rate is below 5, the average D_F will be over 100ms. The experiment results of RT in next sub-section will show that such kind of delay will affect user perceived gaming experience, which we do not want happen in the derivation of I_C.

The impairment function I_C in Table III indicates how much the impairment of VConfig affects the user perceived quality. From Table III, we find that all three games are not sensitive to changes in frame rate from 25 to 15. When frame rate is below 15, game PES is more sensitive to the frame rate than the other two games. For example, the impairment of VConfig (I_C) in game PES jumps over 40 at the frame rate of 8, while it is still around 30 in game WoW and NFS at the same frame rate. Regarding to the resolution changes, game WoW demands high resolution as the value of I_C increases dramatically (and hence MGUE suffers) while the resolution is reduced from VGA to QVGA. However, it seems that quality of the game NFS and PES are not affected significantly by the video resolution reduced from VGA to QVGA.

III.C. Deriving Impairment Functions I_R, I_P, and I_L

To determine I_R, we use the results from the subjective tests, where only RT is varied and all the other factors are kept at their best values.

As expected, the GMOS score goes down when the RT increases in all three games. As an example, figure 5 shows the GMOS scoring by the study group for the game WoW. For each game genre, we define two delay points, T_1 and T_2. T_1 denotes the RT when the GMOS score starts to decrease below 4.5 (R=100), and T_2 denotes the RT where GMOS hits 3.1(R=60). Then T_1 and T_2 divide RT into three

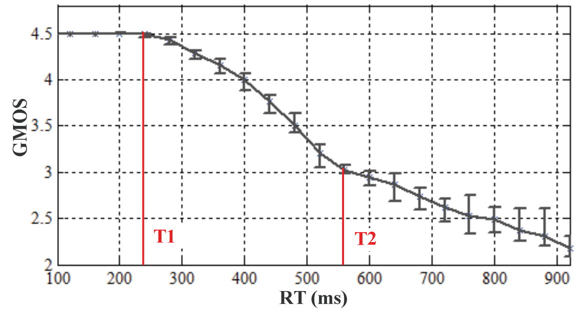

Fig. 5. Subjective test results of RT versus GMOS for WoW

segments. In the first segment (T_1> RT), GMOS keeps at a constant value of 4.5, which implies the user experience remains unimpaired. Therefore, for this segment, the value of impairment function I_R should be 0. In the second segment (T_2> RT >T_1), GMOS decreases from highest 4.5 to 3.1 (minimum acceptable GMOS), while R-factor decreases from 100 to 60, which implies the value of I_R increases from 0 to 40. After T_2, the value of I_R keeps increasing from 40 with a slower slope, denoted by α. We have tried several function models for regression analysis to derive the function of I_R. Based on the experiment results of regression analysis, the linear function as shown in (8) is a simple and accurate model.

$$I_R = \begin{cases} 0 & (T_1 > RT > 0) \\ 40 \times [(RT - T_1)/(T_2 - T_1)] & (T_2 > RT > T_1) \\ 40 + \alpha \times (RT - T_2) & (RT > T_2) \end{cases} \quad (8).$$

Similarly, to derive the impairment function I_P, we analyze the GMOS scores of the corresponding subjective tests (where only PSNR is varied, keeping other factors at their best values). We notice similar trends like displayed by the Delay factor, except that the GMOS score increases while increasing PSNR. We derive I_P as:

$$I_P = \begin{cases} 40 + \beta \times (P_1 - PSNR) & (P_1 > PSNR > 0) \\ 40 \times [(P_2 - PSNR)/(P_2 - P_1)] & (P_2 > PSNR > P_1) \\ 0 & (PSNR > P_2) \end{cases} \quad (9).$$

From the subjective tests corresponding to PLoss, we see a different trend on how it affects user experience. We notice that even low packet loss rate tends to affect the quality and smoothness of the video received by the end device and perceived by the subject. And increasing PLoss will lead to a continuous drop in GMOS. Similar to the derivation of I_R and I_P, we have tried several function models for regression analysis to derive the function of I_L. Based on the experiment results of regression analysis, the linear function (10) is an accurate model to estimate the effect caused by packet Loss:

TABLE IV. The Value of Variables in Equations (8) to (10) For Three Different Games

Game	α	β	γ	T_2	T_1	P_2	P_1
WoW	0.05	5	8	560	240	34	30
NFS	0.08	6	13.5	440	200	33	29
PES	0.12	9	20	360	200	36	33

$$I_L = \gamma \times PLoss \qquad (10).$$

The values of T_1, T_2, P_1, P_2, and coefficients α, β and γ, shown in Table IV, are determined by applying (8) (9) (10) to the subjective test results. This completes the derivation of the impairment functions in (7).

Equations (5) - (10), together with the Tables III-IV, provides the complete MGUE model, which can be used to quantitatively measure the quality of gaming experience over mobile wireless networks using the CMG approach. Note that the values in Tables III and IV apply to the three game genres considered in this study. However, they can be easily extended to other game genres by repeating the approach (subjective study, regression analysis) outlined in this section.

IV. MGUE Prototype, and Model Validation and Enhancement

In this section, we first introduce a software MGUE prototype for measuring the factors and calculating the corresponding GMOS score during live CMG sessions. Next, we validate the accuracy of the MGUE model by conducting another set of subjective experiments. Then we go on enhancing the accuracy of MGUE model with considering cross-effects of impairment factors.

IV.A. MGUE Prototype: Measuring Impairment Factors and Calculating GMOS

We have developed a client-server software MGUE prototype to automatically measure the objective factors (RT, PSNR, and PLoss) during a mobile gaming session, and calculate the corresponding GMOS score, using the MGUE model.

We first design a network probing mechanism that can help CMG server to obtain the client delay D_C and measure server queuing delay D_S, network RDelay, and network PLoss. The CMG server periodically sends a UDP probe to the mobile client (5 probes a second), which includes the probe send out time and probe sequence number. Once mobile client receives a probe, it puts the information of its buffering and decoding delay D_C, into the received probe, and sends it back to the server through the TCP connection. The difference of probe send out time and receive time can indicate the current network round-trip delay RDelay and server queuing delay D_S. And the packet loss rate PLoss can be calculated by checking the received probe sequence number.

Simultaneously, we let CMG server measure the source video PSNR, rendering delay D_R, encoding delay D_E, and frame interval D_F. With the RDelay, D_S, D_C obtained from the network probing mechanism, and D_R, D_F, D_E measured by CMG server, we can calculate the response time RT at the server side (as we have all the parameters in equation (2)).

The above design allows CMG server to real-time simultaneously obtain RT, PSNR, and PLoss. And with the additional information of VConfig, CMG server is able to calculate GMOS score in real time during a CMG session, by using (5) - (10), and the values of I_C (Table III) and T_1, T_2, P_1, P_2, α, β, γ (Table IV) as appropriate for the type of Game being played. In the MGUE prototype, the client need to measure its decoding delay which can be obtained directly from the video player installed on the mobile client. Besides, the client will just forward back the probe packets sent from server. Therefore, the computing load added by MGUE prototype is extremely small (less than 0.01 percent per our test), such that it can be neglected.

IV.B. MGUE Model Validation

Whereas the impairment functions of MGUE model is estimated using test results where only one factor is varying, the accuracy of the MGUE model needs to be validated by conducting another set of controlled experiments considering the effect of simultaneously varying all the factors. We use the same experimental framework as deriving the impairment functions, but a different study group consisting of 15 participants. As opposed to testing the effect of individual factors, we conduct 267 subjective tests where all the factors are varying

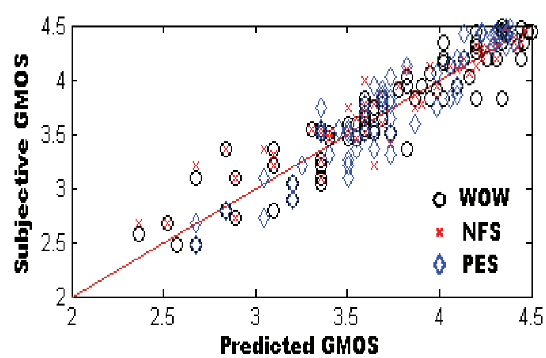

Fig. 6. Relationship between predicted and subjective GMOS

randomly. It should be noted that during some of the validation tests, we also vary server and client utilization to introduce server and client delay. These tests will demonstrate that the way we estimate the impairment function of RT and the RT relation equation (2) are correct. Figure 6 shows the relationship between the MGUE scores predicted by the MGUE model (x-axis) and the subjective (average) GMOS score by the participants (y-axis). The correlation between predicted and subjective user experience is 0.917. This result demonstrates the accuracy of our MGUE model in quantitatively measuring the Mobile Gaming User Experience of a user, given the Game, video configure VConfig used, video PSNR, and RT and Packet Loss experienced during the gaming session.

IV.C. Enhancing MGUE Model by Cross-effect Functions

The above validation results demonstrate that MGUE model derived in section II and III has a good accuracy in predicating user perceived gaming quality. However, our further analysis on validation results discovers that the MGUE model developed in section II and III could perfectly predict GMOS when we only vary one factor but it is not very accurate in predicting GMOS if two or more factors are varied at the same time. Figure 7(a)(b) present the differences of validation results of game WoW between varying one factor and varying several factors. The correlation of predicted and subjective GMOS is 0.965 in figure 7(a), while it is only 0.887 in figure 7(b). Though the overall correlation is good (0.918) as shown in figure 7(c), we believe the accuracy of MGUE can be potentially enhanced if we could address the problem in predicting GMOS when several factors vary simultaneously.

The main reason for the above problem is that we are not taking into account the cross-effects of impairment factors when deriving equation (6). Obviously, the overall impairment of several factors is not exactly the same as the total of the individual impairments of all the factors, though it is related to them. For example, the impairment of 300ms RT and 2% PLoss is different (worse or better) than the total of individual impairment of 300ms RT and individual impairment of 2% PLoss. This difference is distinct especially when individual impairments of two or more factor are significant. To improve our MGUE model, we need to develop cross-effect functions and add them to equation (6). Equation (11) below is the enhanced equation for R-factor considering cross-effects of impairment factors:

$$R = 100 - \sum_i I_i - \sum_{i \neq j} f_{ij}(I_i, I_j) - \sum_{i \neq j \neq k} g_{ijk}(I_i, I_j, I_k) \; - \ldots \; (100 > R > 0) \qquad (11),$$

where $f_{ij}(I_i, I_j)$ denotes the cross-effects of two impairment factors, while $g_{ijk}(I_i, I_j, I_k)$ denotes

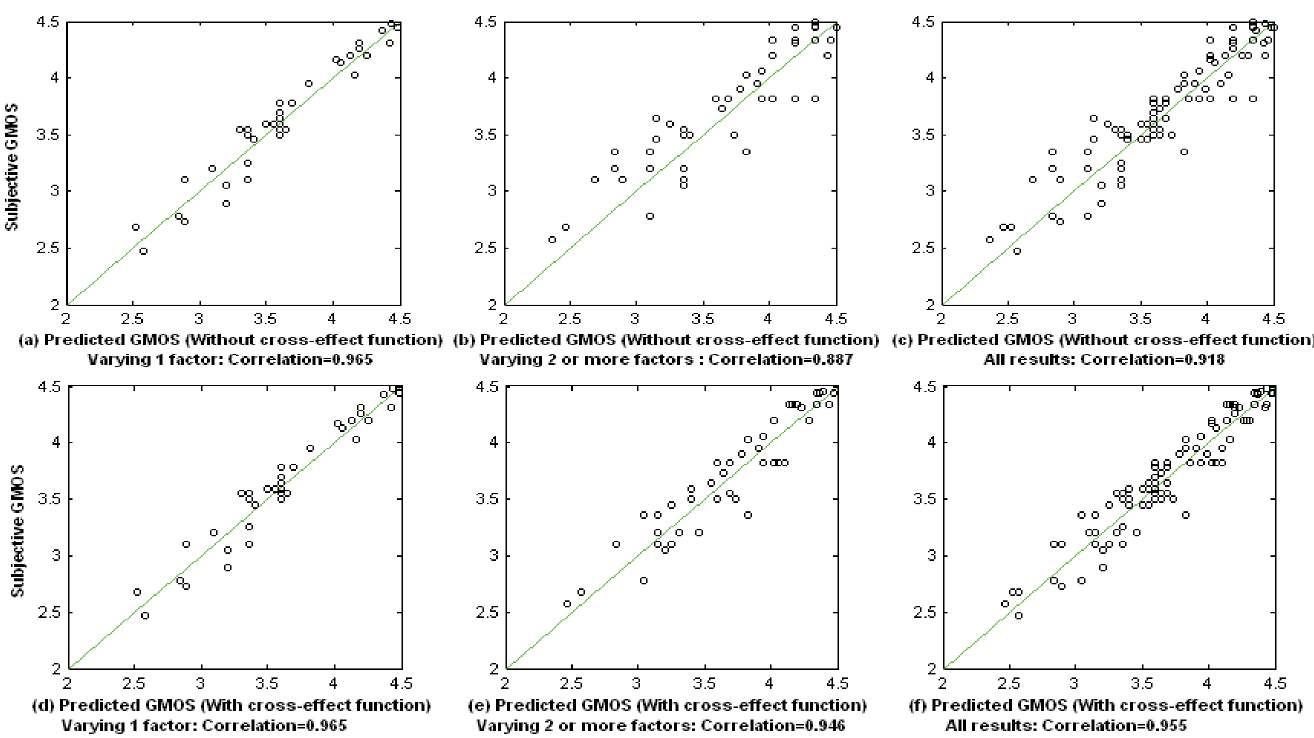

Fig. 7. Correlation of predicted and subjective GMOS: (a) (b) (c) without cross-effect functions; (d) (e) (f) with cross-effect functions

cross-effects of three impairment factors. More the impairment factors used, more cross-effect functions equation (11) will have. However, the MGUE model would be very complex if we consider all kinds of cross-effect functions. Considering the feasibility, we decide to use only pairwise cross-effect functions $f_{ij}(I_i, I_j)$. Thus the enhanced equation for R-factor is:

$$R = 100 - \sum_i I_i - \sum_{i \neq j} f_{ij}(I_i, I_j) \ (100 > R > 0) \quad (12).$$

We have trained the model for R (equation 12) with many different types of functions for $f_{ij}(I_i, I_j)$, including $(I_i + a \times I_j)^n$, $I_i^n \times I_j^m$, and $e^{(I_i + a \times I_j)^n}$. We finally selected the function below (equation 13), because the predicted GMOS scores using this function have the highest correlation with subjective scores obtained from our validation experiments:

$$f_{ij}(I_i, I_j) = k_{ij}\sqrt{I_i \times I_j} \quad (13).$$

The coefficients k_{ij} can be determined by applying equation (13) to the validation test results. As a result, equations (14) (15) (16) are the enhanced equations of R-factor for game WoW, NFS, and PES respectively:

$$\begin{aligned} R = 100 - \sum_{i=1}^{n} I_i &+ 0.05\sqrt{I_C \times I_R} - 0.42\sqrt{I_C \times I_P} \\ &+ 0.10\sqrt{I_C \times I_L} + 0.08\sqrt{I_R \times I_P} + 0.15\sqrt{I_R \times I_L} \\ &+ 0.67\sqrt{I_P \times I_L} \end{aligned} \quad (14).$$

$$\begin{aligned} R = 100 - \sum_{i=1}^{n} I_i &+ 0.05\sqrt{I_C \times I_R} - 0.30\sqrt{I_C \times I_P} \\ &+ 0.36\sqrt{I_C \times I_L} + 0.06\sqrt{I_R \times I_P} + 0.13\sqrt{I_R \times I_L} \\ &+ 0.76\sqrt{I_P \times I_L} \end{aligned} \quad (15).$$

$$\begin{aligned} R = 100 - \sum_{i=1}^{n} I_i &+ 0.02\sqrt{I_C \times I_R} - 0.24\sqrt{I_C \times I_P} \\ &+ 0.12\sqrt{I_C \times I_L} - 0.11\sqrt{I_R \times I_P} + 0.34\sqrt{I_R \times I_L} \\ &+ 0.62\sqrt{I_P \times I_L} \end{aligned} \quad (16).$$

Figure 7(d)(e)(f) present the prediction results of using enhanced equation (14) for game WoW. The correlation of predicted and subjective GMOS has been greatly improved, from 0.887 (figure 7(b)) to 0.946 (figure 7(e)), for the tests where two or more factors are varied. This leads to an improved correlation of total validation results which reaches as high as 0.955 (figure 7(f)), as opposed to only 0.918 (figure 7(c)) without using cross-effect functions.

V. Measuring MGUE in a Mobile Cellular Network

We have also applied enhanced MGUE model (section IV.C) and MGUE prototype (section IV.A) to measure MGUE in real wireless mobile networks. This will also help assess the feasibility and challenges of employing cloud mobile gaming to deliver desired gaming experience in today's mobile wireless networks. The experiments are conducted with a commercially available mobile cellular network. The game server is located in the UCSD campus, while the mobile game is played on a mobile device in four different scenarios: outdoor locations, indoor locations with poor coverage (low Carrier Interference Noise Ratio (CINR)), mobility conditions, and the conditions when cloud server is over-utilized. All three games (genres) and various video settings are used during the testing. Figure 8 shows a representative sample of the data (RT, PLoss, PSNR, and GMOS score) collected from numerous gaming sessions in the mobile cellular network in each of the four scenarios. We make the following observations from our experiments:

1) In outdoor locations, we stream the gaming video at 600kbps data rate, as this is sufficient to ensure very good source video quality (PSNR). The experiments are conducted in both midnight and noon during a day. As presented in figure 8(a), CMG approach can provide the minimum acceptable MGUE (GMOS>3.0) but not very good MGUE (GMOS>4.0) at most times. This is mainly due to the high response time caused by round-trip network delay. As the gaming server is not located in mobile network system, gaming video has to be delivered via multiple hops and routes, including core network backhaul, carrier Ethernet, and wireless access link. Moreover, the gaming commands need to be sent from the mobile device to the server. Therefore, it is challenging to satisfy low response time, and thereby achieve high QoE, using current CMG approach and network architecture. It might be worthwhile to investigate the approaches to reduce this round-trip delay by deploying the gaming servers in the wireless carrier network close to the base stations, as opposed to in the Internet cloud. From the results presented in figure 8(a), we also notice that the user perceived GMOS varies during a day. For example, the average response time during noon is higher than during less busy midnight, leading to a relatively low GMOS score. In addition, the GMOS score during noon has occasional drops

to less than 2.0 at times due to the severe packet losses experienced occasionally.

2) In indoor and in mobility conditions, we test with the gaming video streaming at two data rates: 600kbps with high source video quality as the outdoor test, and 400kbps with reduced source video quality. Figures 8(b) and (c) present sample results tested in both data rates. When we stream the gaming video at high data rate (600kbps), CMG approach cannot provide stable MGUE in both conditions, though GMOS score can reach above 3.0 during brief periods in the mobility conditions. This unacceptable user experience is mainly caused by limited network bandwidth in the indoor and mobility conditions, where the wireless RF link quality can be poor or unstable at times. Unexpected network delay and packet loss will happen, as the wireless channel cannot provide adequate bandwidth for high data rate video streaming application. In contrast, when we stream the gaming video at 400kbps, though video quality is degraded a little at this data rate, the MGUE is more stable and overall better than the MGUE in high data rate (600kbps). The above results show the ability of our MGUE model to capture the networking impairments during cloud mobile gaming sessions on a live mobile wireless network, and provide feedback on the overall quality of user experience, considering the tradeoffs between source video quality impairments, and network impairments. The ability to quantitatively measure the overall user experience under different conditions may help in developing techniques to effect the right tradeoffs between different objective factors and improve user experience during cloud mobile gaming sessions.

3) We also conducted experiments to understand the effect of server utilization/overloading on the user experience during cloud mobile gaming sessions using our MGUE model. As we mentioned before, when cloud gaming server is over-utilized, the rendering delay D_R and encoding delay D_E will increase. In the meanwhile, the generated frame rate will dramatically decrease, which will affect I_C (Table III) and D_F, thus leading to a deteriorating MGUE. As shown in figure 8(d), the CMG approach cannot provide acceptable MGUE when the server computation resource is over-utilized, mainly due to the high response time and low frame rate. Like in the case of networking artifacts, the above results show the ability of our MGUE model to capture server related impairments in cloud mobile gaming user experience, which can be potentially used in the future to develop techniques to appropriately schedule gaming sessions to cloud servers to minimize the server delays and reductions in rendering frame rates.

VI. Conclusions and Future Work

In this article, we address the issue of enabling rich Internet gaming experience on mobile devices, which has the potential of creating a significant new revenue opportunity for mobile service providers and network operators. To alleviate the constraints of current mobile gaming, we discuss the use of a

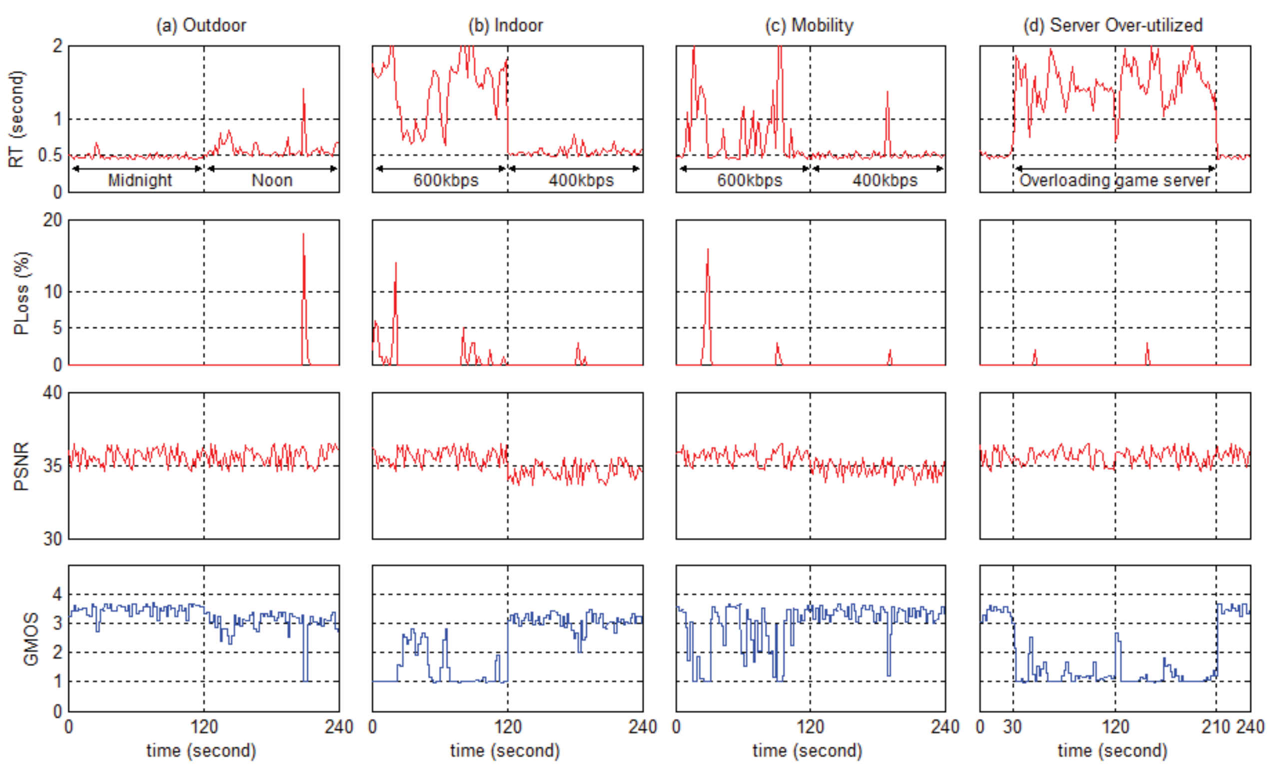

Fig. 8. Test results for game WoW with video settings [VGA and 15 frames per second] in a mobile cellular network

Cloud Mobile Gaming (CMG) approach, and analyze the factors that may affect the Mobile Gaming User Experience (MGUE) using this approach. We develop and validate a MGUE model, to quantitatively measure user experience in this approach, and develop a MGUE prototype measurement tool to enable in-network monitoring of MGUE. We measure and analyze the performance of the CMG approach in a mobile cellular network. Our analysis shows that while it is possible to achieve good quality gaming experience over mobile wireless networks, there are several network conditions in which MGUE may be unacceptable. We suggest future investigation of possible solutions that may lead to significant improvement in user experience achieved by the CMG approach, thus leading to the feasibility of rich, multi-player, Internet games on mobile devices.

The MGUE model discussed in this article can be potentially used by researchers to assess the performance of new CMG techniques, and by future mobile gaming service providers, including network operators, to better plan and optimize their CMG services, as well as monitor in-network the user experience of their mobile gaming subscribers. We also believe that the approach used to develop the MGUE model outlined in this article can be useful for developing QoE models for other new cloud server based interactive multimedia applications, such as virtual reality and augmented reality.

References

[1] D. S. Hands, “A Basic Multimedia Quality Model,” IEEE Transactions on Multimedia, vol. 6, no. 6, pp. 806-816, Dec. 2004.

[2] A. Takahashi, H. Yoshino, and N. Kitawaki, “Perceptual QoS Assessment Technologies for VoIP,” IEEE Communications Magazine, pp. 28- 34, Jul. 2004.

[3] S. Tasaka and Y. Watanabe, “Real-Time Estimation of User-Level QoS in Audio-Video IP Transmission by Using Temporal and Spatial Quality,” IEEE GLOBECOM, Nov. 2007.

[4] K. Yamagishi and T. Hayashi, “Opinion Model for Estimating Video Quality of Videophone Services,” IEEE GLOBECOM, Nov. 2006.

[5] M. Ries, O. Nemethova, M. Rupp, ”Video Quality Estimation for Mobile H.264/AVC Video Streaming,” Journal of Communications, vol. 3, pp. 41 - 50, 2008.

[6] S. Wang, S. Dey, “Modeling and Characterizing User Experience in a Cloud Server Based Mobile Gaming Approach,” in IEEE GLOBECOM, Nov. 2009.

[7] S. Wang, S. Dey, “Rendering Adaptation to Address Communication and Computation Constraints in Cloud Mobile Gaming,” in IEEE GLOBECOM, Dec. 2010.

[8] M. Bredel, M. Fidler, “A Measurement Study regarding Quality of Service and its Impact on Multiplayer Online Games”, In ACM SIGCOMM workshop on Network and System Support for Games, (NetGames), 2010.

[9] M. Claypool and K. Claypool, “Latency and Player Actions in Online Games,” Communications of the ACM, 49(11), 2006.

[10] M. Dick, O. Wellnitz, and L. Wolf, “Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games”, In ACM SIGCOMM workshop on Network and System Support for Games, (NetGames), 2005.

[11] W.-C. Feng, F. Chang, W.-C.Feng, and J. Walpole, “A Traffic Characterization of Popular On-line Games”, IEEE/ACM Tranactions on Networking, vol 13, no.3, pp.588-500, Jun. 2005.

[12] T. Fritsch, H. Ritter and J. Schiller, “The Effect of Latency and Network Limitations on MMORPGs - (A Field Study of Everquest2)”, In ACM SIGCOMM workshop on Network and System Support for Games, (NetGames), 2005.

[13] T. Yasui, et al., “Influences of Network Latency and Packet Loss on Consistency in Networked Racing Games,” In ACM SIGCOMM workshop on Network and System Support for Games, (NetGames), 2005.

[14] ITU-T Recommendation ITU-T P.800, “Methods for Subjective Determination of Transmission Quality,” 1996.

[15] ITU-T Recommendation G.107, “The E-model, a Computational Model for Use in Transmission Planning,” Mar. 2005.

[16] ITU-T J.144, “Objective Perceptual Video Quality Measurement Techniques for Digital Cable Television in the Presence of a Full Reference,” Mar. 2004.

[17] ITU-T Contribution COM12-D13-E, “Proposal on Basic Concepts of a Multimedia Quality Assessment Model,” Jan. 2005.

[18] ITU-T Recommendation G.1070, “Opinion Model for Videotelephony,” Apr. 2007.

[19] A. Jurgelionis et al., “Platform for Distributed 3D Gaming”, International Journal of Computer Games Technology, Volume 2009, Article ID 231863.

[20] I. Nave, H. David, A. Shani, A. Laikari, P. Eisert, and P.Fechteler, “Games@Large Graphics Streaming Architecture,” Proceedings of the 12th Annual IEEE International Symposium on Consumer Electronics, pp. 1–4, Algarve, Portugal, Apr. 2008.

[21] “Onlive”, http://www.onlive.com.

Reviews and Authors' Response for paper: Cloud Mobile Gaming: Modeling and Measuring User Experience in Mobile Wireless Networks

Reviews

The paper discusses a model for quantifying user gaming experience in a setup where the game execution is performed on a remote cloud server. The model consists of different factors that can affect the user gaming experience in such a setup: packet losses, response times, server delay etc. I like the topic and the overall problem that is trying to be addressed in this paper. The paper is also written well.

Strengths:

- The paper presents an interesting topic and does good job in identifying a set of certain key metrics that can easily captured to quantify the gaming experience of the player in a cloud mobile gaming setup. It uses these metrics to create a model to quantify the user gaming experience. Such a model would be very useful for game developers to keep track of the gaming experience of the players and understand the extent of the impact of the different factors that can degrade the experience of the players.
- The paper evaluates the model across different game genres in real world scenarios.

Weaknesses:

- The instrumentation should be described more precisely, since it is used to capture the various metrics that feed into the model. For example, what is the frequency of the probing mechanism? What is the overhead of the measurements performed at the client?
- The paper mentions that the model in a game genre should be similar. This is a strong statement to make given that the paper hasn't experimented with multiple games in the same genre and the evaluation uses the same game. So, it still seems that one would have to perform custom measurements for every game to generate an accurate model. It would be very interesting to have an evaluation section that uses a different game from the same genre than the one used to create the model in order to understand how accurately the model holds within each genre.
- The model requires some effort to compute the impairment factor for each metric individually and to take care of the dependencies between the factors the model goes upto using pairs of factors. Thus, it requires some careful effort to capture the impact of the various metrics by controlling some factors and varying the others. How does it compare against machine learning techniques such as using SVM to train a model by varying the different factors in random proportions without the need to carefully control the various factors? If a simple SVM model performs nearly well, then it cuts down the effort needed to create such a model. Such a discussion is important to motivate the need for a more carefully constructed model.

A model to capture the user gaming experience in real time is useful but this work need some more evaluation to address the concerns discussed before.

Please find below some of the concerns we have about this paper.

The equation for GMOS calculation comes out of the blue. While it is mentioned that the authors have duplicated the function for E-model, we are not sure why GMOS calculation should be similar to the subject metric used in audio transmission systems. More so, we are not convinced about the linear model used for impairments. Also, what happens when the value of R becomes negative? Is the minimum clamped to zero (in the presence of multiple impairments). Specifically, it seems to me that all the impairments are given equal weights. Why is this so? We would assume the effect of one impairment could be much more than the other, depending on the type of game.

It seems to us that the model parameters are very specific to the games being played, and the conditions/settings under which they are played. What happens when any of these are changed? For example, if we were to play a new game and use GMOS metric to measure the MGUE, would I have to re-train the entire model using 100 odd participants? That seems like a lot of overhead.

We do not understand the contribution of experiments in Figure 8 quite as clearly. It only shows that as each of the impairment comes into effect, a corresponding (and expected) response is shown by the GMOS metric. What is the experiment trying to demonstrate here? Put another way, if we come up with an entirely new equation that essentially captures MGUE as a function that is inversely proportional to the impairments, wouldn't we see similar graphs? What then is the contribution of these experiments? We would have loved to see experiments, with real people, in real settings (playing mobile games on a cellular network), let them "rate" their experience, and then benchmark the performance of measured GMOS.

Improving the model by taking into the account the cross-effects of impairments was certainly the right thing to address/improve the model. I was hoping this would take care of my concern about the "weights" given to the impairments and how they interact with one another. However, again, the authors seem to have come up with a magic formula to address these effects without any justification behind the functions used ("after trying several functions" is not enough an explanation.).

Authors' Response

To address the concerns of the reviews, the paper has been enhanced in various ways. Following are some of the main response comments from the authors.

We have added more description about the probing mechanism and the overhead on client in section IV.A.

We have not claimed that a single MGUE model for a genre will serve for each game of that genre. The contribution of our work is to derive a general MGUE modeling framework which can be used for all kinds of cloud mobile games. The metrics of MGUE in Table I and coefficients in equation 13 might be different from game to game, and will need to be derived using the steps outlined in the paper, including the subjective assessment experiments, for each different game. We have performed measurements for three different games belonging to three different genres to highlight the differences different impairment factors will have on games belonging to different genres, but to a much lesser degree, the impairments may be different for different games belonging to the same genre also.

Though machine learning techniques like SVM can be potentially used to train a MGUE model from the experimental data we have collected, its accuracy will be questionable. To maximize the accuracy of the impairment factors, we have made certain observations from the experimental data, for example, that response time will only affect user experience when it is above a certain threshold, and used these observations to develop the impairment factors. As a result, our derived model can achieve a high correlation (of predicted and subjective GMOS) of about 0.95 as reported in the paper. To address the reviewer's comment, we conducted a preliminary study using a popular SVM tool, LIBSVM, to train a model for R (equation 7) using the experimental data in section IV.B. Using the resulting R model (equation 7) given by LIBSVM, the correlation of predicated GMOS (calculated by equations 6 and 7) and subjective GMOS (section IV.B) is under 0.8. This validates the need for the approach proposed in this paper to achieve a more accurate MGUE model.

Though E-model [15] is originally proposed for audio transmission system, it has been used for video application also. We choose this E-model because it provides a good statistical mapping in converting the quality rating factor R into a MOS scale. Rating factor R ranges from 0-100. If the result of equation 6 is less than 0, the value of R is 0. We have added this clarification to the paper. We have derived an enhanced non-linear MGUE model, where the coefficients of different impairments are not same (equations 14, 15, and 16). Also, please note that the coefficients are different for different games, showing that the impairment factors will affect user experience in different manner for different games.

The reviewer is right, that for each game the metrics of MGUE model in table I and coefficients in Equation 13 need to be trained by a group of subjective assessment experiments. However, we do not feel that this will be a significant overhead for a game, considering the amount of effort and money usually spent to develop a typical Internet game. More importantly, such kind of training experiments will have to be conducted only once, but the model derived for the game can be used forever.

This paper has two contributions. The first one is deriving a MGUE model to measure mobile gaming user experience. The second one is to apply the MGUE prototype on real networks to characterize and assess the feasibility of Cloud Mobile Gaming in terms of user experience achieved. Figure 8 presents experimental results for the second contribution, highlighting the feasibility and challenges of Cloud Mobile Gaming to achieve acceptable user experience in real network conditions.

We have explained in details the steps of how we add the cross-effects of impairments to our model in the paper. Firstly, we give equation 11 as a comprehensive model which considers the cross-effects of impairments. Then we simplify equation 11 to equation 12 to reduce complexity of the model. To derive functions $f_{ij}(I_i, I_j)$ in equation 12, we have trained with many different types of functions, including $(I_i + a \times I_j)^n$, $I_i^n \times I_j^m$, and $e^{(I_i + a \times I_j)^n}$. We finally selected $\sqrt{I_i \times I_j}$ (equation 13), because the predicted GMOS scores using this function have the highest correlation with subjective scores obtained from our validation experiments. We have added the above clarification to the paper.