Performance Measurements on a Cloud VR Gaming Platform

Yen-Chun Li
Department of Computer Science
National Tsing-Hua University
Hsin-Chu, Taiwan

Chia-Hsin Hsu
Department of Computer Science
National Tsing-Hua University
Hsin-Chu, Taiwan

Yu-Chun Lin
Department of Computer Science
National Tsing-Hua University
Hsin-Chu, Taiwan

Cheng-Hsin Hsu
Department of Computer Science
National Tsing-Hua University
Hsin-Chu, Taiwan

ABSTRACT

As cloud gaming and Virtual Reality (VR) games become popular in the game industry, game developers engage in these fields to boost their sales. Because cloud gaming possesses the merit of lifting computation loads from client devices to servers, it solves the high resource consumption issue of VR games on regular clients. However, it is important to know where is the bottleneck of the cloud VR gaming platform and how can it be improved in the future. In this paper, we conduct extensive experiments on the state-of-the-art cloud VR gaming platform–Air Light VR (ALVR). In particular, we analyze the performance of ALVR using both Quality-of-Service and Quality-of-Experience metrics. Our experiments reveal that latency (up to 90 ms RTT) has less influence on user experience compared to bandwidth limitation (as small as 35 Mbps) and packet loss rate (as high as 8%) . Moreover, we find that VR gamers can hardly notice the difference between the gaming experience with different latency values (between 0 and 90 ms RTT). Such findings shed some lights on how to further improve the cloud VR gaming platform, e.g., a budget of up to 90 ms RTT may be used to absorb network dynamics when bandwidth is insufficient.

CCS CONCEPTS

• **Information systems** → **Multimedia streaming**; • **Human-centered computing** → **Virtual reality**.

KEYWORDS

Measurement; Virtual Reality; Computer Games; Cloud Computing; Prototype

ACM Reference Format:

Yen-Chun Li, Chia-Hsin Hsu, Yu-Chun Lin, and Cheng-Hsin Hsu. 2020. Performance Measurements on a Cloud VR Gaming Platform. In *1st Workshop on Quality of Experience (QoE) in Visual Multimedia Applications (QoEVMA'20), October 16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3423328.3423497

ACM ISBN 978-1-4503-8158-1/20/10…$15.00
https://doi.org/10.1145/3423328.3423497

1 INTRODUCTION

Cloud gaming refers to rendering visually-appealing game scenes on powerful cloud servers and streaming the game scenes to weak client devices. Cloud gaming has been increasingly more popular, as indicated by multiple research reports. For example, it is pointed out that the annual global cloud gaming market will grow from USD 306 million (2019) to 3.17 billion (2022) [4]. Indeed, companies like Sony, Nvidia, Apple, and Google have realized the business potential of cloud gaming and released their cloud gaming services. These services enable gamers to play games on different networked devices, such as laptops and mobile phones that were not powerful enough to execute the games themselves. With such services, games can be quickly loaded, paused, and resumed anytime and anywhere. This allows gamers to play their favorite games across all their devices.

Virtual Reality (VR) games attract more and more attention because of the popularity of commodity Head-Mounted Displays (HMDs). A market research report projects that the VR annual market will grow from USD 78 million (2019) to 1.37 billion (2025) [14]. Compared to traditional computer games, offering high-quality cloud VR gaming experience is more difficult, because: (i) VR games consume much more resources and (ii) VR gamers demand much shorter response time and higher visual quality. The feasibility of executing VR games on cloud gaming platforms, however, has not been thoroughly investigated.

In this paper, we take a very first step to understand the performance of cloud VR gaming platform by conducting both the objective and subjective measurements. In particular, we build a VR gaming testbed on an open-source project, called Air Light Virtual Reality (ALVR) [10]. We set up a Windows workstation as the ALVR server and an Android HMD as the ALVR client. The server and client are interconnected by a BSD workstation as the traffic shaper, which allows us to emulate different network conditions by injecting propagation delay, packet loss rate, and bandwidth limitation. By instrumenting our testbed, we compare the objective performance of the cloud-based VR games under different network conditions. Besides, we also recruit gamers to play VR games

This work was partially supported by the Ministry of Science and Technology of Taiwan (#107-2221-E-007-091-MY3).

and provide their inputs, so as to compare the subjective performance of the cloud-based VR games under different network conditions. To the best of our knowledge, the performance of different VR games on a cloud gaming platform has never been studied in the literature. Our measurement studies reveal several novel findings.

2 RELATED WORK

Huang et al. [8] set up a cloud gaming testbed similar to ours, in order to measure the cloud gaming performance under different network conditions. For the most relevant metric: latency, they propose to divide the total latency into several components and conduct component-wise measurements. As follow-up work, Huang et al. [9] focus on the performance measurements on resource-constrained mobile clients of a cloud gaming testbed. Various parameters, such as the resolution, bitrate, and frame rate, are varied in their study. Ryan et al. [5] also measure the interactive latency and image quality of cloud games in their experiments. They conduct the same measurements on different game types, so as to understand the implications of game types on the cloud gaming performance. More similar measurement studies can be found in cloud gaming surveys [5] and references therein. Xue et al. [18] carry out a measurement study on multiple cloud gaming platforms. The authors measure three metrics, which are the bitrate, packet rate, and latency, to understand the interplay between the platform architecture and performance. Michael et al. [11] perform a subjective user study on cloud gaming. They identify the Key Influence Factors (KIF) for Quality of Experience (QoE) using their propose measurement environment and procedure. In particular, they vary the propagation delay and packet loss rate, while collecting user inputs for computing the Mean Opinion Score (MOS). These studies [5, 8, 9, 11, 18] concern general computer games, while our current paper focuses on VR games.

Zhong et al. [20] presents a wireless VR platform for commodity HMDs. The authors use a programmable WiGig interface to replace the Ethernet cable between the desktop server and HMD client. By doing so, they attain high flexibility to reconfigure the network link to maximize VR gaming performance in response to dynamic network conditions. They consider two performance metrics: (i) frame-processing time by instrumenting a VR game and (ii) end-to-end latency by shooting videos from two HMDs with synchronized high-speed cameras. By comparing the performance with WiGig and Ethernet, they quantify the implication of using a wireless link to replace the wired one. Liu et al. [13] also proposes a wireless VR platform for HMDs using WiGig. They pipeline the encoding, transmission, and decoding stages to reduce the end-to-end latency. To evaluate their platform, they measure several metrics: (i) end-to-end latency, (ii) visual quality, (iii) frame loss rate, and (iv) client resource usage. Unlike our current paper, these two papers [13, 20] do not focus on performance measurements of cloud VR gaming platforms. In contrast, Chang et al. [6] measure the performance of an HMD client. In particular, they set up a testbed to quantify the timing and position accuracy of the HMD. However, they do not offload the rendering and game logics to a cloud server.

Some other papers in the literature move towards higher levels and study the image/video quality of 360-degree content. For example, Sun et al. [16] survey on state-of-art image quality assessment approaches on compressed 360-degree images. They implement 18 image quality metrics and also investigate the subjective quality. In another work [17] from the same authors, they develop a Convolution Neural Network (CNN) model for evaluating 360-degree image quality. In contrast to traditional metrics, they show their model performs better. Duan et al. [7] focus on perceptual 360-degree image quality assessment. They take view direction and eye movement into considerations in the evaluations. Yao et al. [19] classify 360-degree videos into four classes on temporal and spatial complexities. Moreover, they adopt the Analysis of Variance (ANOVA) test to identify the key factors. They then propose QoE models base on these important QoE factors. While the above studies [7, 16, 17, 19] consider a different usage scenario than VR gaming, the lessons learned in them can be useful when developing application-specific quality metrics for cloud VR games.

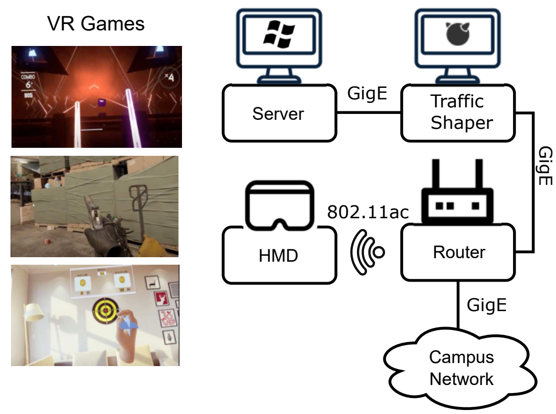


Figure 1: Our cloud VR gaming testbed.

3 TESTBED IMPLEMENTATION

In this section, we present our VR cloud gaming testbed.

3.1 Setup

As illustrated in Fig. 1, we have implemented a VR cloud gaming testbed with four hardware devices: (i) a Windows workstation with an Intel Core i5 CPU, 64 GB RAM, Nvidia RTX 2080Ti GPU serving as the cloud gaming server, (ii) a BSD workstation with Intel i5 CPU serving as the traffic shaper, (iii) an ASUS RT-AC88U 802.11 ac WiFi router that supports 40-MHz channels at 5 GHz bands, and (iv) an Oculus Quest HMD, which runs Android 7.1.1 OS on a SnapDragon 835 CPU with 4 GB RAM, Adreno 540 GPU, and 802.11 ac network interface. The HMD supports up to 1440 × 1600 per-eye resolution at 72 frame-per-second (fps). It also tracks eye gazes and lip movements. Fig. 4(a) gives a photo of our testbed in the lab. We can see the server desktop, traffic shaper, router, and client HMD constitute our testbed.

The ALVR project consists of two entities: an ALVR server and an ALVR client. We run both the ALVR server and SteamVR [3]

Table 1: Spatial and Temporal Complexity

Game	Spatial Info.	Temporal Info.
Beat Saber	70.85	36.18
Half-Life-Alyx	45.06	18.18
Together VR	49.36	8.07

on our cloud server. The ALVR server adopts the OpenVR APIs to extract the VR gaming scenes from SteamVR in RGB and depth videos. OpenVR is a close-sourced API developed by Valve for VR game developers. Upon getting the game scenes, the ALVR server employs DirectX calls to compress the videos and send the video frames to the AVLR client by UDP. The ALVR client receives the video frames and calls OpenGL ES APIs [2] to draw the VR viewports in the HMD. The ALVR server also sends the depth images to the ALVR client, in order to generate the stereoscopic HMD viewports (for left/right eyes). In a reverse channel, the control signals are sent over TCP for reliable transmission. To emulate the diverse and dynamic network conditions, we install Dummynet [1] on the BSD workstation, which allows us to adjust propagation delay, packet loss rate, and network bandwidth.

3.2 Game Selection

We carefully selected the following three representative VR games for our experiments.

- **Beat Saber** is a rhythm VR music game. In this game, players slash the boxes, which are flying toward them, on the beats. We chose Beat Saber because it is a latency-sensitive game with faster pace. We conjecture that its gaming experience is highly affected by the latency.
- **Half-Life-Alyx** is an open-world VR adventure game. In this game, players shoot zombies and gather loot in the virtual world. We chose Half-Life-Alyx because it's a popular first-person shooting game among VR gamers. It provides vivid gaming experience, but is resource demanding.
- **Together VR** is a leisure interactive game. In this game, players experience everyday life with a female avatar. We chose Together VR because it has the slowest pace with simple game scenes.

To understand the characteristics of these three VR games, we adopt the Spatial Information (SI) and Temporal Information (TI) [15] to quantify their complexity. We record each gameplay for 2 mins, and then calculate its SI and TI values. Table 1 reports the results, which shows that: (i) Beat Saber is more complex than Half-Life-Alyx in both spatial and temporal domains and (ii) Together VR has the lowest temporal complexity than other games and comparable spatial complexity to Half-Life-Alyx. We also measure the resource consumption of these three VR games in the next section. Among the three games, Half-Life-Alyx is more resource-hungry than others.

4 OBJECTIVE MEASUREMENT STUDY

In this section, we conduct several experiments while emulating diverse network conditions to understand how network conditions affect the objective performance of our cloud VR gaming testbed.

4.1 Design

We measure the following metrics.

- **Frame rate.** The successfully received frames by the ALVR client divided by the time duration.
- **Latency**. The total latency from capturing to rendering of each video frame. It can be further divided into encoding, transport, and decoding latency.
- **Server load (CPU/GPU).** The CPU/GPU (cycle) usage and CPU/GPU memory usage.
- **Client load (CPU/GPU).** The CPU/GPU (cycle) usage.
- **Throughput**. The actual data size received by the ALVR client divided by the time duration.

We note that the latency and frame rate are measured by instrumenting the ALVR source code. For metrics that require matching video frames at the server and those at the client, we adopt DirectX 11 API to overlay timestamp and frame index on the captured video frames at the ALVR server before sending them to the ALVR client. We adopt several Python libraries (psutil, GPUtil, and pynvml, to name a few) to measure the server load, and Android ADB tool to measure the client load and throughput. All measurements are saved into log files with a sample rate of 1 Hz if not otherwise specified.

To emulate realistic network conditions, we vary the following parameters throughout our measurements.

- **Bandwidth.** We cap the bandwidth among {35, 40, 45, 50, 100} Mbps. As we will see later, 100 Mbps is much higher than the encoding bitrate. Thus, it essentially represents no bandwidth limitation.
- **Delay.** We vary the one-way delay among {0, 5, 15, 25, 35, 45} ms. We inject the same delay in both directions.
- **Packet loss rate.** We vary the packet loss rate among {0%, 2%, 4%, 6%, 8%}. We drop packets in both directions.

These parameters are enforced by the Dummynet on the BSD-based traffic shaper. We have found that 30 Mbps encoding rate is sufficient to carry good enough video quality for cloud VR games through some pilot tests. Hence, we configure the ALVR server to encode the captured videos at 30 Mbps in our measurement study. In total, we consider 14 different combinations of the network parameters. We measure each VR games three times. Each measurement lasts for 3 minutes, and we filter out the measurement results in the first and last minutes. Hence, for each pair of network conditions and VR games, we got three 1-min measurement results. We note that, ALVR chooses HEVC as its default video encoder. If Nvidia GPU is available, ALVR uses NVENC as the encoder. Hence, in our experiments, ALVR server captures the SteamVR gaming scenes, encodes them with NVENC encoder, and renders it on Oculus Quest at 2880×1660 resolution and 72 fps for each eye.

4.2 Results

We first plot sample measurement results on Beat Saber in Fig. 2. We only adjust one network parameter at a time. For instance, when varying the bandwidth, we do not add additional delay nor inject packet losses. We make several observations on this figure.

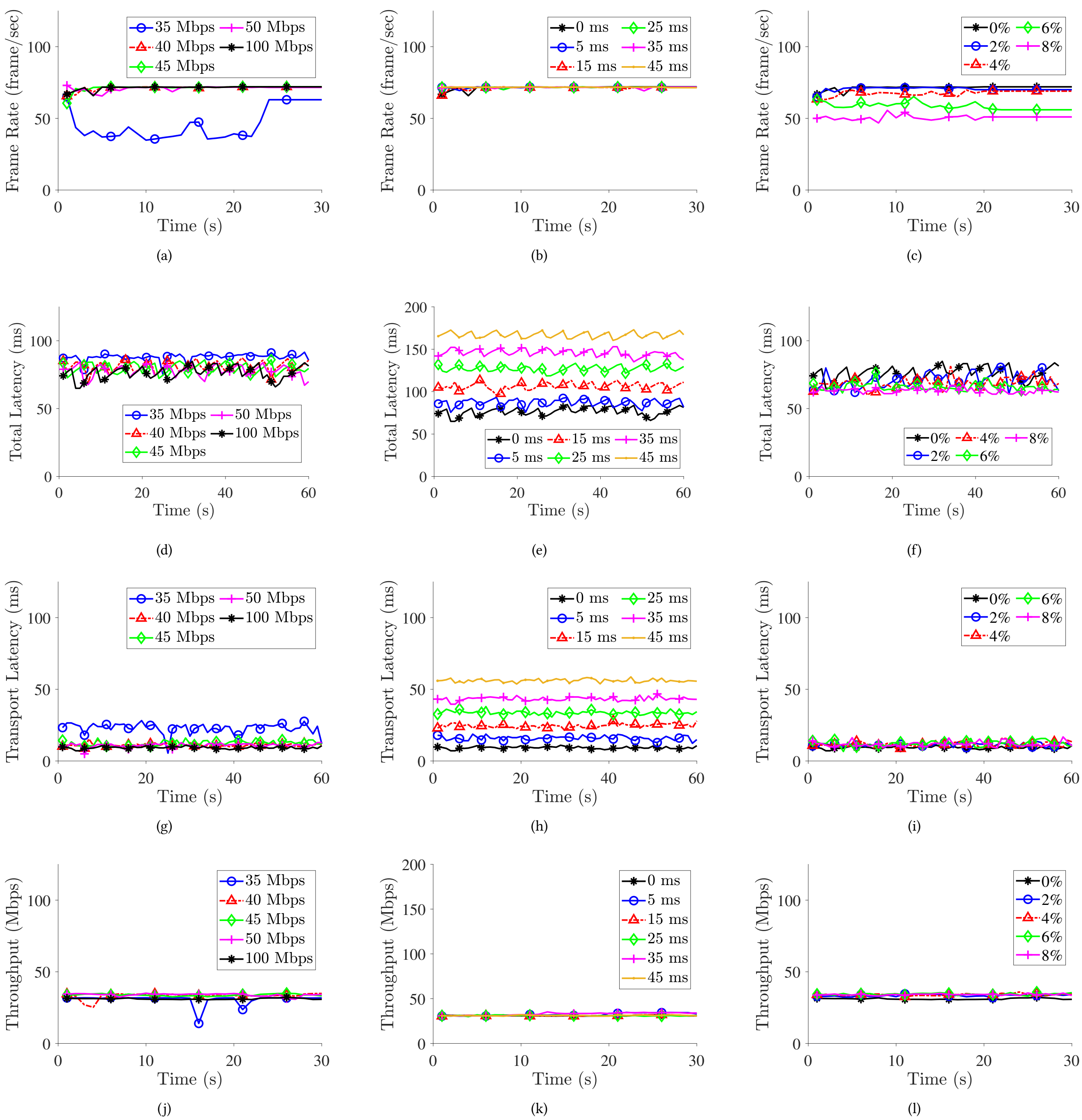


Figure 2: A sample run of Beat Saber. Results on: frame rate (a)–(c); total latency (d)–(f); transport latency (g)–(i); throughput (j)–(l) under diverse network conditions: bandwidth (a), (d), (g), (j); delay (b), (e), (h), (k); packet loss rate (c), (f), (i), (l).

- **Frame rate is sensitive to insufficient bandwidth and nontrivial packet loss rate (Figs. 2(a)–2(c)).** Fig. 2(a) reveals that when the network bandwidth is capped at 35 Mbps, we start to see drops on the frame rate. Compared to the encoding bitrate of 30 Mbps, we conclude that the protocol and control overhead accounts for at least 5/35 = 14.29% of the bandwidth consumption, if not higher. Fig. 2(c) shows that the frame rate drops when the packet loss rate

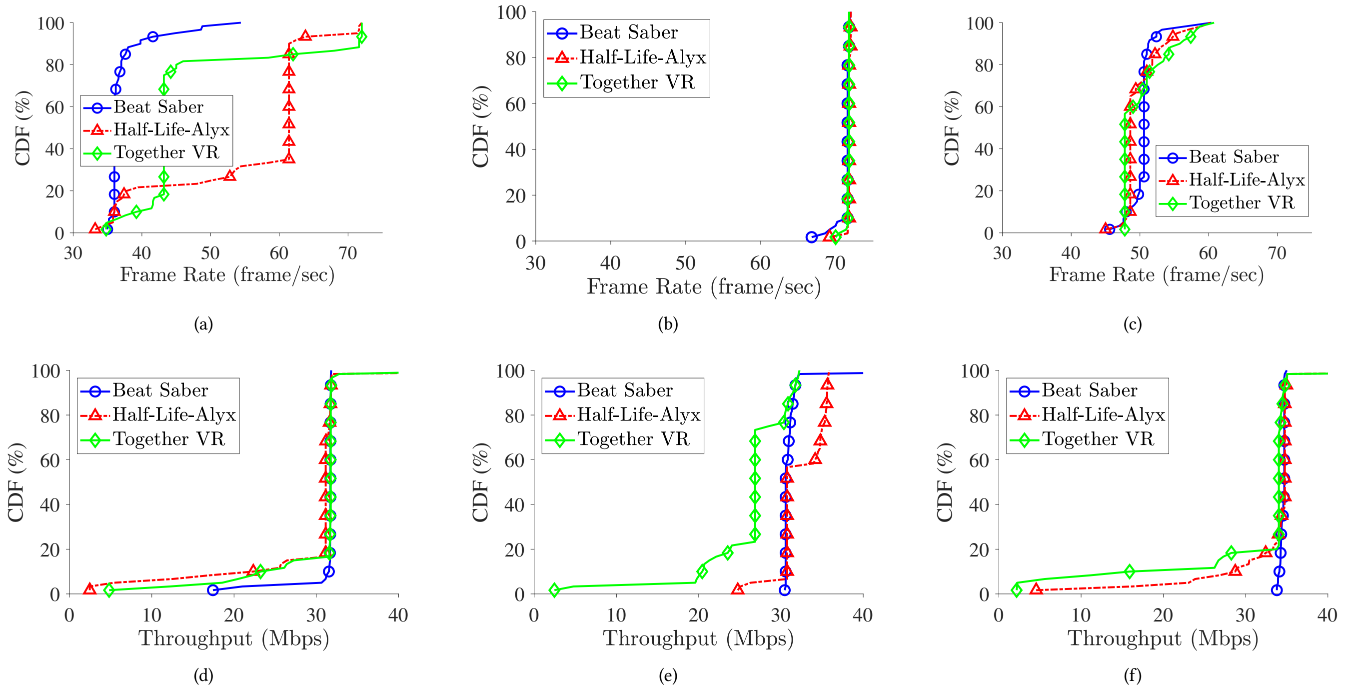


Figure 3: Overall CDF curves off frame rate (a)–(c); throughput (d)–(f) under challenging network condition: bandwidth of 35 Mbps (a), (d); delay of 45 ms (b), (e); packet loss rate of 8% (c), (f).

Table 2: Resource Consumption of Different VR Games

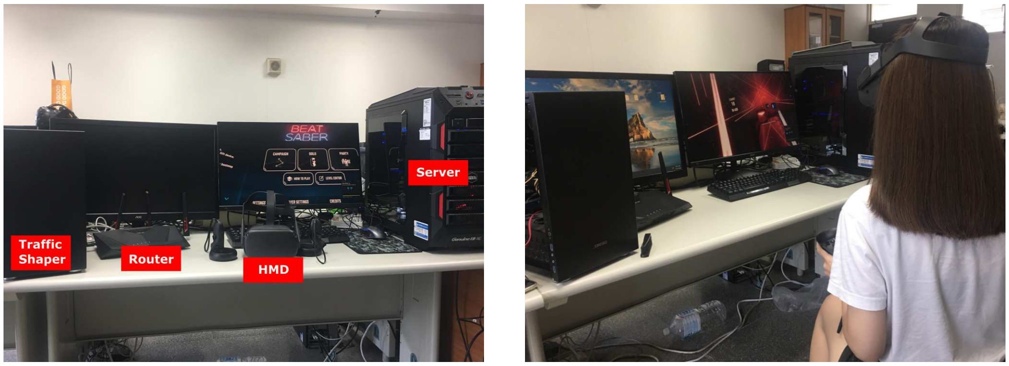
	Server CPU (%)	MEM (GB)	GPU (%)	GPU MEM (%)	Client CPU (%)	GPU(%)
			Beat Saber			
Mean	32.73	6.47	17.38	24.47	43.80	40.48
Std.	1.99	0.01	0.52	0.30	25.86	0.91
			Half-Life-Alyx			
Mean	43.68	8.47	42.4	49.50	11.80	38.93
Std.	11.06	1.05	10.37	10.02	3.68	6.51
			Together VR			
Mean	31.95	7.01	28.03	29.48	12.71	32.77
Std.	3.77	0.11	9.45	3.05	4.60	5.80

is at 4+%. This may be attributed to the UDP-based streaming adopted by our VR cloud gaming testbed and the error correction mechanism implemented by the decoder.

- **Total latency is affected by all three network parameters (Figs. 2(d)–2(i)).** While the relation between injected delay and measured total latency is clearly shown in Fig. 2(e), the total latency is also affected by the bandwidth (Fig. 2(d)) and the packet loss rate (Fig. 2(f)). More precisely, lower capped bandwidth results in more congested networks, and thus longer total delay (Fig. 2(d)). This can be observed by the higher transport delay in Fig. 2(g). Moreover, the total latency drops when the packet loss rate increases (Fig. 2(f)). This is, in fact, a side effect of lower frame rate: fewer frames lead to lower client CPU/GPU load. Indeed, our measured client CPU/GPU usages drops from about 30%/40% to about 10%/30% when the packet loss rate increases from 0% to 8%. The figures are not shown due to space constraints.
- **Throughput is not affected by different network conditions (Figs. 2(j)–2(l)).** Our VR cloud gaming testbed consumes about 35 Mbps bandwidth under all conditions. This may be explained by the properties of its UDP-based streaming, which is more aggressive on using network resources.

Next, we aggregate the measurement results from all runs and compare the performance of different VR games. We plot the Cumulative Distribution Functions (CDFs) under challenging network conditions in Fig. 3. We also report the overall resource consumption across all measurements in Table 2. We make the following observations.

- **Different game implementations could affect the performance of our cloud VR gaming testbed (Fig. 3(a),**



(a) (b)

Figure 4: The user study is conducted on: (a) our cloud VR testbed with (b) subjects playing VR games.

Fig. 3(e), and Table 2). Intuitively, the streaming performance of cloud VR gaming testbeds is *orthogonal* to the VR game implementation, as after the game scenes are captured, they are all treated as regular videos. This, however, is not the case in our measurement study. For example, Fig. 3(a) reveals that Half-Life-Alyx retains the highest frame rate under the bandwidth constraint of 35 Mbps, while Beat Saber suffers the most in the same condition. Compared to the target frame rate of 72 fps, more than half of the time, Beat Saber suffers from sub 37 fps frame rate, while Together VR suffers from sub 42 fps and Half-Life-Alyx only suffers from sub 60 fps. Similarly, Fig. 3(e) also depicts that Half-Life-Alyx generates the highest network throughput, while Beat Saber generates the least. The higher frame rate and higher throughput of Half-Life-Alyx are inline with the numbers reported in Table 2: Half-Life-Alyx consumes more server resources in general as it generates more frames even when the network resources are scarce. Unfortunately, our selected VR games are all *proprietary*, which prevents us from studying how they invoke the SteamVR APIs. A deeper investigation, perhaps with open-source VR games, is among our future work.

- **Both server and client are over-subscribed in terms of CPU and GPU resources (Table 2).** The table depicts that both the server and client are resourceful for running the VR games and ALVR software. At the server-side, Half-Life-Alyx leads to the highest resource consumption, which can be attributed to its higher frame rate under constrained network conditions. On the client-side, the Beat Saber results in the highest resource consumption. We believe this is because of its higher input rates and temporal changes (as shown in Table 1). Nonetheless, our VR cloud gaming testbed is powerful enough for running the selected VR games.

5 SUBJECTIVE MEASUREMENT STUDY

In this section, we study how diverse network conditions affect user experience.

5.1 User Study Design

We conduct a user study to quantify the user-perceived Quality-of-Experience (QoE). We recruit ten subjects from our department. The subjects are between 21 to 30 years old. Seven of them are male, and seven of them have played VR games before. None of our subjects suffer from vision illness nor severe motion sickness. Fig. 4 shows a subject playing VR games on our testbed. First, we ask our subjects some basic information and help them put on the HMD. Then we give each subject a training session so as to make him/her more familiar with playing VR games in HMDs. The games used in the training session is the same with the games in the actual user study. Next, we ask them to play VR games using our testbed. Similar to the objective measurement study, we change the delay, packet loss rate, and bandwidth limitation to emulate diverse network conditions. For each game, there are 9 different network conditions[2]. With three VR games, we divide each subject's test into 27 game sessions. The order of 27 game sessions is random. When a subject finishes a game session, he/she takes a one-minute break. Meanwhile, we ask the subject for the QoE scores and gamer performance (i.e., game scores), before continuing the next session. The QoE scores range from 1 (lowest) to 5 (highest). The QoE scores are on the following questions: *image quality*, *fluency*, *immersion*, *continue playing*, *sickness*, and *latency*. Among them, for the last two questions, lower QoE scores are better; for other questions, higher are better.

5.2 Results

We compute the MOS scores of different questions along with the 95% confidence intervals across the subjects. Figs. 5 and 6 give the detailed MOS scores, where the former (latter) figure reports all questions in which higher (lower) MOS scores are better. The MOS scores of different VR games are quite differentiable, while the network conditions clearly affect the user experience. We made the following three sample observations.

- **Gamers of Beat Saber are more sensitive to delay compared to those of the other two games.**

[2]We limit the number of network conditions to avoid fatigue of the subjects.

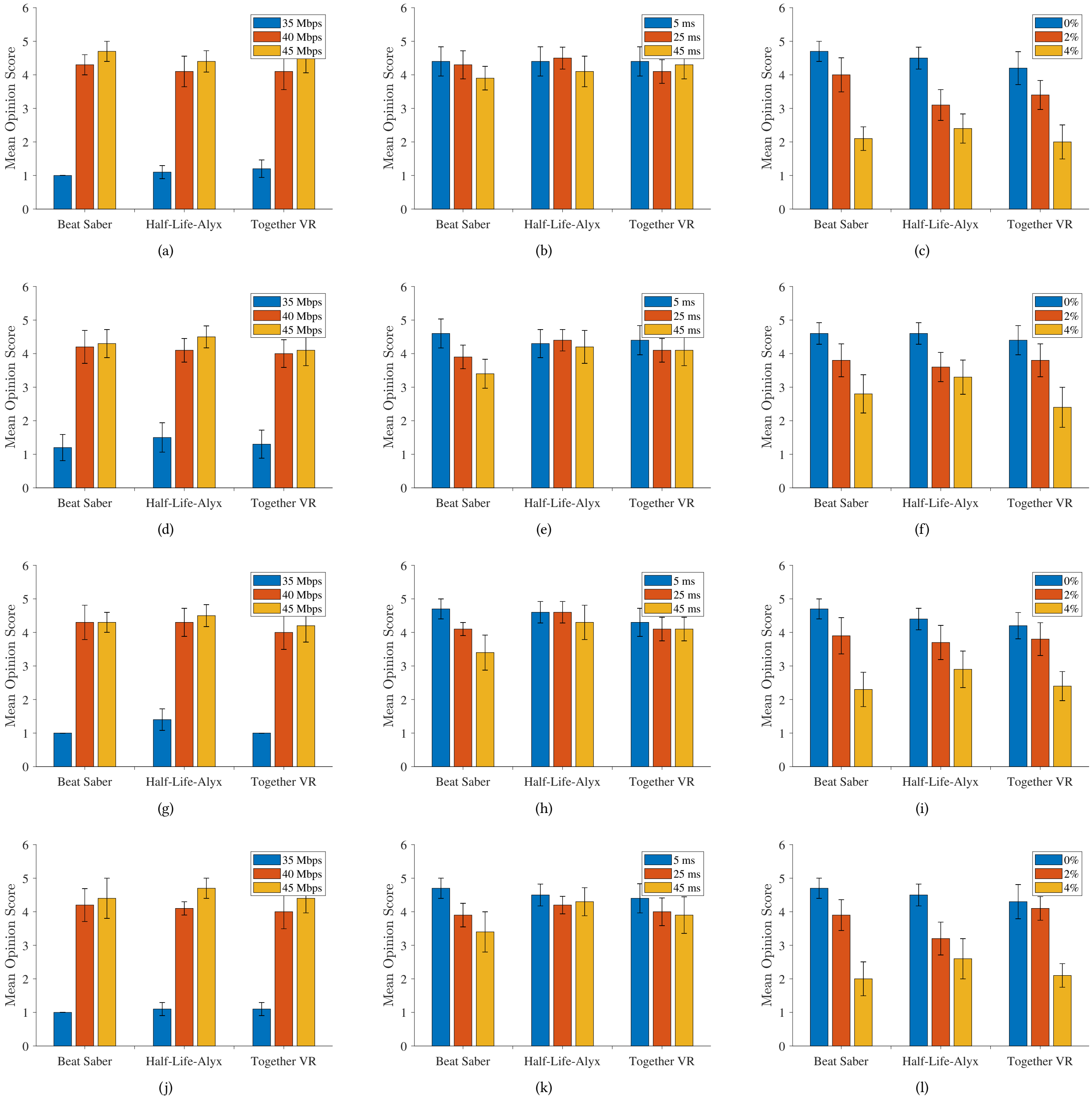


Figure 5: MOS scores of different VR games. Image quality: (a)–(c); fluency: (d)–(f); immersion: (g)–(i); continue playing: (j)–(l) under diverse network conditions: bandwidth (a), (d), (g), (j); delay (b), (e), (h), (k); packet loss rate (c), (f), (i), (l). Higher MOS scores mean better performance in these questions.

- **Gaming experience significantly drops once the bandwidth goes below 35 Mbps.**
- **More gamers are turned away when the packet loss rate goes up to 4%.**

Note that we do not point the readers to specific subfigures that support the above observations, because the trends are readily seen. Besides, many additional observations can be made by the readers from the two figures. Overall, the MOS scores observed in

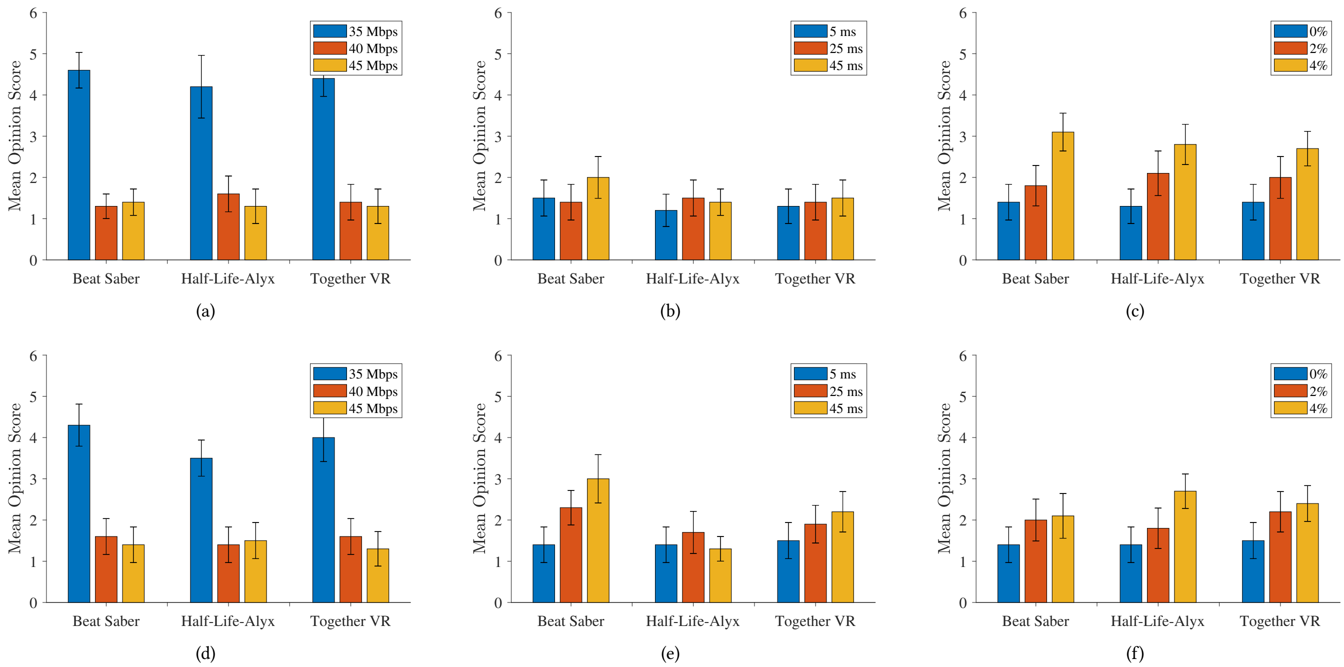


Figure 6: MOS scores of different VR games. Sickness: (a)–(c); latency: (d)–(f) under diverse network conditions: bandwidth (a), (d); delay (b), (e); packet loss rate (c), (f). Lower MOS scores mean better performance in these questions.

our subjective measurement study are inline with the results of our objective measurement study.

6 CONCLUSION AND OUTLOOKS

We set up a testbed with a traffic controller to evaluate the performance of the state-of-the-art cloud VR gaming testbed under different network conditions. In particular, we conduct both objective and subjective measurement studies. We made quite a few observations on the measurement results. For example, within the considered ranges of bandwidth limitation, delay, and packet loss rate, we find that the delay has less influence on user experience. In contrast, when limiting the bandwidth or increasing the packet loss rate, gamers suffer from extensive blocking artifacts. Furthermore, compared to Half-Life-Alyx and Together VR, the delay imposes more influence on Beat Saber. This makes sense, because Beat Saber is a rhythm game, in which gamers need to be responsive enough to slash boxes. Readers may also draw new observations on the detailed figures given in the paper.

Our paper can be extended in several directions.

- **Measure the video quality at the HMD client.** Video quality plays an important role on user experience. Therefore, we want to quantify the quality degradation due to imperfect network conditions. We are extending our testbed to achieve that.
- **Employ modern network protocols, like QUIC [12] .** The ALVR project was designed for VR streaming over short-range wireless networks. Therefore, it opts for the raw UDP packets, which may not be suitable for cloud VR gaming over the Internet. For example, raw UDP packets offer zero protection over malicious attackers in the Internet. We plan to integrate modern protocols, such as QUIC with ALVR and quantify the merits enabled by these protocols.
- **Support adaptive VR gaming over dynamic networks.** Our experiments indicate that when the bandwidth limitation is close to the sending rate, the packet loss rate increases, which negative affects the user experience. This is because the current ALVR project only supports static encoding and sending rates. We plan to design, implement, and evaluate bitrate adaptation algorithms for our cloud VR gaming testbed.
- **Optimizing the cloud VR streaming via super resolution algorithms.** The current ALVR project only supports Oculus Quest at 1440×1600 per eye. In the future, the HMD resolutions will increase, which would consume more bandwidth. Super resolution technique may be a solution to reduce the required bandwidth of cloud VR gaming. By reducing the traffic amount, the super resolution algorithms may also reduce the network delay (transmission delay, more specifically). We are currently exploring different super resolution algorithms to find the one that works the best with rendered game scenes. The super resolution algorithms can run on an edge server or on the client.

REFERENCES

[1] 2002. *Dummynet, A Traffic Shaper, Bandwidth Manager and Delay Emulator.* https://www.freebsd.org/cgi/man.cgi?dummynet.

[2] 2016. *OpenGL ES Official Website.* https://www.khronos.org/opengles/.

[3] 2016. *SteamVR Official Website.* https://store.steampowered.com/steamvr.
[4] 2020. *Cloud Gaming Market by Offering, Device Type, Solution, Gamer Type, Region–Global Forecast to 2024.* https://reurl.cc/ld5j89.
[5] Wei Cai, Ryan Shea, Chun-Ying Huang, Kuan-Ta Chen, Jiangchuan Liu, Victor CM Leung, and Cheng-Hsin Hsu. 2016. A Survey on Cloud Gaming: Future of Computer Games. *IEEE Access* 4 (2016), 7605–7620.
[6] Chun-Ming Chang, Cheng-Hsin Hsu, Chih-Fan Hsu, and Kuan-Ta Chen. 2016. Performance Measurements of Virtual Reality Systems: Quantifying the Timing and Positioning Accuracy. In *Proc. of ACM International Conference on Multimedia (MM'16).* Amsterdam, The Netherland, 655–659.
[7] Huiyu Duan, Guangtao Zhai, Xiongkuo Min, Yucheng Zhu, Yi Fang, and Xiaokang Yang. 2018. Perceptual Quality Assessment of Omnidirectional Images. In *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS'18).* Shanghai, China, 1–5.
[8] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen. 2013. GamingAnywhere: An Open Cloud Gaming System. In *Proc. of ACM Multimedia Systems Conference (MMSys'13).* Oslo, Norway, 36–47.
[9] Chun-Ying Huang, Yu-Ling Huang, Yu-Hsuan Chi, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2015. To Cloud or Not to Cloud: Measuring the Performance of Mobile Gaming. In *Proc. of ACM Workshop on Mobile Gaming (MobiGame'15).* Florence, Italy, 19–24.
[10] JackD38. 2017. *ALVR Github.* https://github.com/JackD83/ALVR.
[11] Michael Jarschel, Daniel Schlosser, Sven Scheuring, and Tobias Hoßfeld. 2011. An Evaluation of QoE in Cloud Gaming Based on Subjective Tests. In *Proc. of IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'11).* Seoul, South Korea, 330–335.
[12] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, and Janardhan Iyengar. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *Proc. of International Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'17).* Los Angeles, CA, 183–196.
[13] Luyang Liu, Ruiguang Zhong, Wuyang Zhang, Yunxin Liu, Jiansong Zhang, Lintao Zhang, and Marco Gruteser. 2018. Cutting the CORD: Designing a High-Quality Untethered VR System with Low Latency Remote Rendering. In *Proc. of Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'18).* Munich, Germany, 68–80.
[14] ResearchAndMarkets. 2020. *Global Virtual Reality (VR) Market 2019-2025: Expected to Rise at a CAGR of 32.9%.* https://reurl.cc/e8m40x.
[15] slhck. 2017. *SITI Github.* https://github.com/slhck/siti.
[16] Wei Sun, Ke Gu, Siwei Ma, Wenhan Zhu, Ning Liu, and Guangtao Zhai. 2018. A Large-Scale Compressed 360-Degree Spherical Image Database: From Subjective Quality Evaluation to Objective Model Comparison. In *Proc. of IEEE nternational Workshop on Multimedia Signal Processing (MMSP'18).* Shanghai, China, 1–6.
[17] Wei Sun, Xiongkuo Min, Guangtao Zhai, Ke Gu, Huiyu Duan, and Siwei Ma. 2020. MC360IQA: A Multi-channel CNN for Blind 360-Degree Image Quality Assessment. *IEEE Journal of Selected Topics in Signal Processing* 14, 1 (2020), 64–77.
[18] Zheng Xue, Di Wu, Jian He, Xiaojun Hei, and Yong Liu. 2014. Playing High-End Video Games in the Cloud: A Measurement Study. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 12 (2014), 2013–2025.
[19] Shun-Huai Yao, Ching-Ling Fan, and Cheng-Hsin Hsu. 2019. Towards Quality-of-Experience Models for Watching 360° Videos in Head-Mounted Virtual Reality. In *Proc. of International Conference on Quality of Multimedia Experience (QoMEX'19).* Berlin, Germany, 1–3.
[20] Ruiguang Zhong, Manni Wang, Zijian Chen, Luyang Liu, Yunxin Liu, Jiansong Zhang, Lintao Zhang, and Thomas Moscibroda. 2017. On Building a Programmable Wireless High-Quality Virtual Reality System Using Commodity Hardware. In *Proc. of ACM Asia-Pacific Workshop on Systems (APSys'17).* Mumbai, India, 1–7.