Local and remote fog based trade-offs for QOE in VR applications by using cloudXR and Oculus air link

Gabriele Maiorano
Department of Computer, Control and Management Engineering
"Antonio Ruberti"
Sapienza University of Rome
Rome, Italy
maiorano.1961926@studenti.uniroma1.it

Gabriele Proietti Mattia
Department of Computer, Control and Management Engineering
"Antonio Ruberti"
Sapienza University of Rome
Rome, Italy
proiettimattia@diag.uniroma1.it

Roberto Beraldi
Department of Computer, Control and Management Engineering
"Antonio Ruberti"
Sapienza University of Rome
Rome, Italy
beraldi@diag.uniroma1.it

***Abstract*—Virtual Reality (VR) is now a well-established technology which offers realistic and immersive virtual worlds to the user usually by means of Head-Mounted Displays (HMDs). Actually, these devices can also be backed by a cloud server which can host the game server or even directly render the virtual world, as in the well-known cloud gaming paradigm. However, due to the drastically low latencies that technology requires, it is more convenient, when possible, to use servers that are as close to the users as possible. As a consequence, implementing the game or the render server in the Fog Computing layer is a concrete possibility. In this paper, we investigate, by using the Meta Quest 2 device, which are the QoE trade-offs, in terms of graphic quality and network performance, both in the case in which the HMD performs the 3d rendering locally by using a sample game written with Unreal Engine and in the case in which the 3d rendering is done in the Fog by means of the nVidia CloudXR framework and Oculus Air Link. From the results of our experiments, we found that remote rendering offers a stable frame rate against a higher quality image. Instead, local rendering sets the best possible graphics quality against the optimal frame rate. Additionally, we saw how remote rendering uses video compression in the case of decreasing bandwidth available to adjust graphics quality and FPS. The same does not hold for Motion-to-Photon latency, which increases with distance, reducing the general QoE.**

***Index Terms*—virtual reality, unreal engine, qoe, cloudxr, fog computing**

I. Introduction

The current advance in Internet technologies and standards (e.g. Wi-Fi 6[1] and 5G), along with fiber optics and new generations of routers, allowed users to play games on devices without the usage of powerful graphic cards, e.g. Google Stadia[2]. This new era of gaming is defined as cloud-based gaming. In the case of Virtual Reality, this idea is applied in a scenario where a remote server renders frames of complex and high-demanding application and send them back to an HMD device. Instead, the device will only send input action, like head movement or controller button pressed, to the server to be processed. The usage of Virtual and Augmented Reality technologies is reaching more application from the gaming field to engineering and medical sector thanks to its immersiveness, that captures the attention of the customers in the case of retail industry [1]–[3], or used as support to improve and facilitate work of medical teams in the case of trauma surgeon inspection [4]–[6], but also for therapy and mental stimulation [7].

In order to maintain a stable frame rate at a value of at least 72 frames-per-second (FPS) [8], [9], the majority of the available applications are based on low-demanding graphics rendering settings. This is because lower values of framerate are the main causes of the so-called Visually induced motion sickness (VIMS) [10], [11]. Furthermore, it is important to keep responsiveness between the user's motion and the action reproduced in-game [12], which is another reason for sickness.

The power of cloud gaming mainly relies on remote GPU power and user network strength. In fact, with a good remote rig (e.g. with an NVIDIA RTX 3080Ti), also the last generation AAA games can be enjoyed without requiring to own the same environment.The same idea can be applied to complex VR games, which could express all their graphics power obtaining a more realistic and fascinating virtual experience. However, one of the disadvantages of cloud gaming is the requirement for a stable and high-quality Internet connection, to diminish latencies and reduce the amount of time needed to send back and forth data of high dimensions. For example, Oculus Air Link requires for the network, to ensure the best performance:

- the PC connected to router/access-point via Ethernet cable;
- the router supporting Wi-Fi 5 (802.11AC) or Wi-Fi 6 (802.11AX);
- headset connected to 5GHz Wi-Fi band.

Another solution for cloud-gaming is the one developed

[1]https://docs.broadcom.com/doc/80211ax-WP
[2]https://stadia.google.com

by NVIDIA, called CloudXR[3], an SDK for Extended Reality (XR) which enables the streaming of OpenVR applications over radio signal (Wi-Fi or 5G). In this architecture, the server presents a virtual HMD driver to SteamVR, faking the local connection of the device, and thus not requiring any changes at the application level. The main challenge of this software is to achieve both the highest quality graphics plus mobile freedom. To reach higher QoS, they found a solution to challenges such as latency and Bandwidth variation, using a custom HEVC profile and video compression.

In this paper, we perform different experiments by installing these frameworks in a Fog server that is attached to the same WiFi6 router to which the HMD is connected, thus enabling a fog gaming paradigm. We implemented a sample VR application in different builds where we have inserted visual effects in order to stress the graphic card. Along with it, other parameters to manage were latency and bandwidth at variable distances of the HMD to the router. The main goal of this research is to study if, depending on the quality of the graphics settings, local and remote rendering are equivalent and if, depending on the distance of the device to the router, VR-cloud gaming is still a good option to take into consideration when we talk about Virtual Reality gaming. This paper explores the capabilities of cloud gaming in a Fog environment, trying to find bottlenecks of remote rendering in the case of Virtual Reality with an evaluation of FPS, Bandwidth, Delay and QoE. The rest of the paper is organized as follows. In Section II, we give some information on related works about network performances, Cloud-gaming with a focus also on VR Cloud-gaming. In section III, several experimental setups are introduced with details on the hardware and software used. Section IV shows the results obtained in the previous experiments conducted and Section V briefly resume what we have found with these.

II. Related Work

Several studies about Cloud-Gaming shown the effects of latency on users' Quality of Experience (QoE).

Sabet et al. proposed a latency compensation technique for Cloud-Gaming, showing that Spatial Accuracy, Temporal Accuracy and Predictability are the characteristics that mitigate delays influences [13].

Lampe et al. [14], and previously Chen, Chang, Tseng, Huang and Lei [15], proposed a software tool in order to measure the latencies of the main cloud-gaming providers. The former with GAme LAtency MEasurement Tool (GALAMETO.KOM) which autonomously invokes actions and waits for the time interval until those are represented in-game. The latter instead proposed a methodology to measure latency and applied it on two platforms, OnLive and StreamMyGame. In particular, they used the hooking mechanism to inject code into the clients and measure through that as timestamps: t_0 to define the moment in which they start measurement and t_4 as the timestamp in which the menu screen appears. Therefore they have calculated the response delay as a $\Delta = t_4 - t_0$

Furthermore, Di Domenico, Perna, Trevisan, Vassio and Giordano compared providers (Google Stadia, GeForceNow, and PS Now) on the protocols used and bandwidth spent [16].

Regarding local rendering but in a network environment, other studies focused on non-network parameters, such as FPS [17], [18]. In particular, Lindblom, Laine and Rossi [8] developed MiReBooks, an educational project for mixed reality mining, focused on FPS in a fog environment with a variable number of concurrent-user (CCU), whilst Parthasarathy, Simiscukay, O'Connorz and Muntean made experiments increasing CCU and analyzing performances about bandwidth [19].

Zhao, Allison, Vinnikov and Jennings [20] instead, measured the Motion-to-Photon latency in HMD through the use of a pendulum, introducing a damped sinusoidal motion to the device. Going deeper and focusing only on XR Cloud-Gaming, Liubogoshchev, Ragimova, Lyankhov, Tang and Khorov [21] considered a sample CloudXR architecture, represented with a mathematical model of a discrete state Markov Chain, estimating some parameters of QoE such as network capacity and bitrate adaption function.

Zhou et al. [22] used NVIDIA CloudXR[4] as framework in order to better disseminate heritage, in the case of Augmented Reality (AR).

Similar to our paper research, Li, Chia-Hsin Hsu, Lin and Cheng-Hsin Hsu [23] proposed an experimental setup for measuring latencies and frame rate with three different typologies of VR-games (TogetherVR requiring low-demanding graphics, Half-Life-Alix requiring high-demanding graphics and Beat Saber, which is highly latency-sensitive) in a cloud environment. As result, they found that frame rate is sensitive to insufficient bandwidth and that the gaming experience radically decreases once the bandwidth goes below 35Mbps.

However, none of this article focused on differences between local and remote rendering taking into consideration graphics quality and variable network bandwidth. Moreover, the main limitation of the previous cited existing systems is that they are used for general cloud gaming and not for VR/AR cloud gaming. Instead, NVIDIA CloudXR, which at the time of writing is available as an early access program, do not allow more than one user client per running server.

III. Experimental Setup

A. Equipment

In our experiments, we used a Meta Quest 2[5], a VR Headset equipped with Qualcomm Snapdragon XR2 CPU with support for WiFi6 (802.11ax), 6GB of RAM, an Adreno 650 GPU, an LCD panel display with an 1832×1920 per-eye resolution, which can run at a refresh rate of up to 120 Hz, and as input, 6DOF inside-out tracking through 4 built-in cameras and 2 controllers with accelerometers and gyroscopes.

[3]https://developer.nvidia.com/nvidia-cloudxr-sdk

[4]https://developer.nvidia.com/nvidia-cloudxr-sdk

[5]https://store.facebook.com/it/en/quest/products/quest-2

As server, we used a PC with 32GB RAM, AMD Ryzen 9 5900x CPU and an NVIDIA RTX 3060 GPU, connected to the router via Gigabit Ethernet.

The main application, installed on the HMD and the server is developed through the engine Unreal Engine (UE) at version 4.27[6]. It is a demo version, where a user in VR can interact with another user connected through the PC version of the game passing a big ball. The whole session is managed by the use of the LAN session of UE. In order to stress the graphics, we inserted 120 visual effects (VFX) in the scene, that can be enabled/disabled by pressing a button, 60 lights at a time, as can be seen in Figure 1-2.

To summarize, the possible testing situations w.r.t. the graphic settings, were the following:

- high demanding: 120 VFX;
- medium demanding: 60 VFX;
- low demanding: 0 VFX.

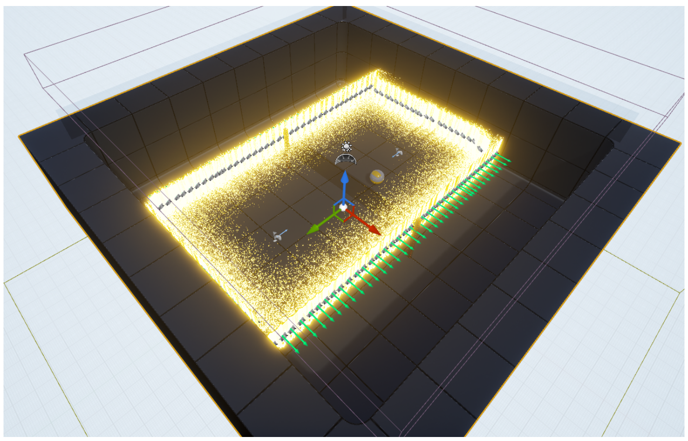

Fig. 1: The Main level of the game developed with Unreal Engine 4.27, with additional 120 visual effects.

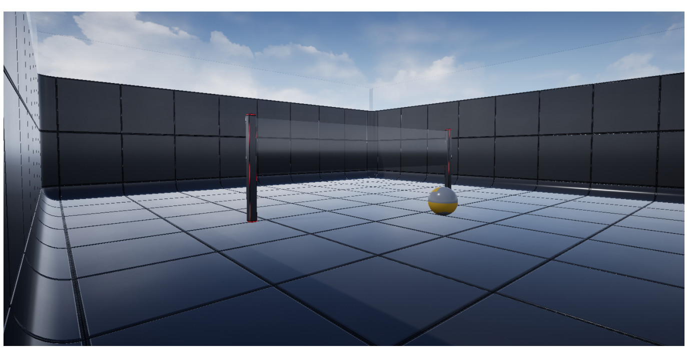

Fig. 2: The Main level of the game developed with Unreal Engine 4.27, without visual effects.

Furthermore, with the goal of testing the worst-case network quality, e.g. too many hosts connected in the same network or large room which will lead to high signal dispersion, we decided to limit network bandwidth to 250 Mbps, from the router's settings, since the bandwidth available was reaching up to 1 Gbps. This also reduced the strength of the radio signal while reaching the end of the path.

B. Local Rendering

In the case of local rendering, the application is installed on the HMD device, and the rendering pipeline is fully managed by it. Furthermore, while running the game, OVR Metrics Tool was recording data[7]. In particular, this tool provides performance information about a running application in a similar way as through the usage of VrApi Logcat logs and, used in report mode, records the performance obtained during a VR session, which can be exported as a CSV. Indeed, we focused on FPS, since the rendering was managed locally on the HMD. Furthermore, because of no constraints on router distance, we performed all the tests without changing position.

C. Remote Rendering

For the remote rendering tests, we used NVIDIA CloudXR, along with SteamVR[8]. In this case, the HMD device decodes the frames rendered on the server, and displays them on the screen. Concerning the profiling of parameters, thanks to Android Studio, we modified the native code of the client version of CloudXR, and then installed it on the Quest 2, for saving on a file, information about FPS and Network parameters. So, we added the Desktop version of the game on Steam library and, after that, we set up the server version of NVIDIA CloudXR on the PC and the client version on the Quest2 as in Figure 3. In this test, we decided to start from a distance very close to the router, and then move along a path without obstacles of about 40 meters and finally go back. The path was followed by taking one step per second at a velocity of approximately 0.67 m/s for a complete test duration of 2 minutes. We repeated this experience with all the different visual settings previously described.

The parameters taken into consideration for the remote rendering are those shown in Table I.

D. Remote Rendering with Oculus AirLink

The main difference between the previous scenario and this one is that Cloud XR is suited for remote and local servers, while Oculus AirLink works only in a LAN setting, where HMD and the PC server are both in the same network. In this situation, we can control the Screen of the computer through the Oculus Quest. One advantage of Oculus AirLink is that there is no limit cap at 72 FPS. All values are recorded with the debugger Oculus Debug Tool[9]. In particular, the parameter we were interested in is the Motion-To-Photon latency (MTP) defined as the time needed for a user movement to be fully reflected on a display screen. It is very important in VR since an MTP value higher than 20ms causes spatial disorientation and dizziness.

[6]https://www.unrealengine.com

[7]https://developer.oculus.com/documentation/native/android/ts-ovrmetricstool/

[8]https://steamvr.com/

[9]https://developer.oculus.com/documentation/native/pc/dg-debug-tool/

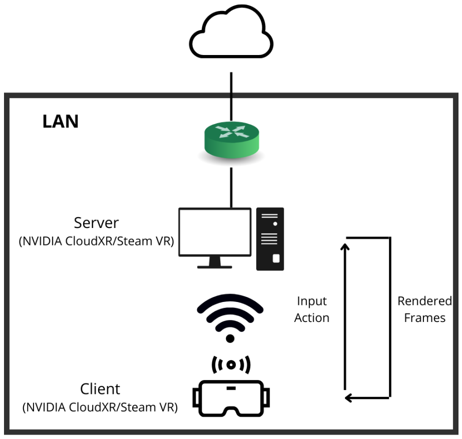

Fig. 3: Architecture of the LAN setting using NVIDIA CloudXR for Remote rendering.

IV. Results

As previously said, all the cases of local and remote rendering were tested with different graphics settings. In particular, in the graph that will follow the 3 cases can be distinguished by the following colours:

- Blue Line: High Demanding (120 VFX)
- Orange Line: Medium Demanding (60 VFX)
- Green Line: Low Demanding (0 VFX)

A. Local Rendering

1) Frames (Figure 4): With respect to FPS, only without VFX is possible to reach a mean value of 72 FPS required to better enjoy the experience. On the contrary with all the VFX active, the values obtained are not optimal for a VR scenario, with an average value of 41 FPS. With the case in the middle, we reach a mean value of 55 FPS which also is not good for VR gaming. The drops present in the graph are due to the movement of the motion controllers and of the HMD which are added to the process of rendering.

B. Remote Rendering with NVIDIA CloudXR

1) Frames (Figure 5): In the case of Remote Rendering, we reached optimal performance with respect to no VFX. The FPS to be rendered were higher than 72, but limited to that value because of the constraint of CloudXR. Instead, FPSs were lower, but not so bad in the case of all active VFX. In particular, the curve in the orange and blue case are following a sawtooth-like pattern, and, with the goal of trying to keep constant that value, while moving along the path, the rendered quality was decreasing according to NVIDIA CloudXR video compression. Furthermore, we argue that the several peaks reached in the blue and orange case are the exact moment in which the video stream has lowered the graphics quality. The other three parameters taken into consideration for delays are due to:

- FDT: end-to-end frame transmission;
- FQT: frames waiting in the rendering queue;
- FLT: frames waiting to be latched in the process of video-encoding.

We found that the general delay (FDT) has on average a value of 50 ms, and at the maximum distance it reaches a value of 150 ms. The same trend happens for FQT. Instead, in the last graph of Figure 5 (FLT), only in the simple case we have a constant trend with a mean value of 0.15 ms.

2) Bandwidth (Figure 6): As for the previous parameters, also information about bandwidth is retrieved through the native code of the client application. In particular, Bandwidth utilization displayed in figure refers to the average video rate registered every second, while Bandwidth available is an estimation of the transmission capacity also that recorded every second and depending on the network availability. During the experiment, the Bandwidth available from the original value of 250 Mbps, after the first 15 m starts falling in all the 3 cases. Furthermore, whilst the simple case required less throughput

TABLE I: Parameters taken in consideration during remote rendering experiment, divided in the following categories: Frames, Network Bandwidth, Delay and Quality

Symbol	Name	Meaning
FPS	Frames Per Second	The number of frames consecutively displayed each second
FDT	Frame Delivery Time	The average time between a frame being submitted on the server and getting latched and released on the client
FQT	Frame Queue Time	The average time a frame spends queued on the client before it is latched and released
FLT	Frame Latch Time	The average time the client application spends waiting for a frame to be latched
BA	Bandwidth Available	The estimated available bandwidth from server to client
BU	Bandwidth Utilization	The average video streaming rate from server to client
RTD	Round Trip Delay	The estimated network round trip delay between server and client
QoE	Quality of Experience	The measure of the delight or annoyance of the user of an application with a 1-to-5 rating from bad to excellent

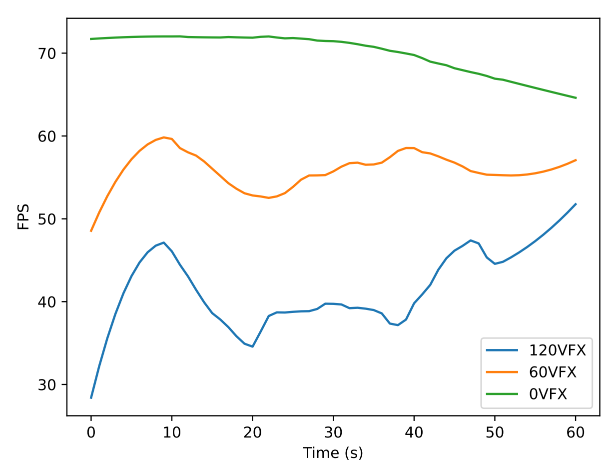

Fig. 4: Behaviour of the FPS for the local rendering experiment.

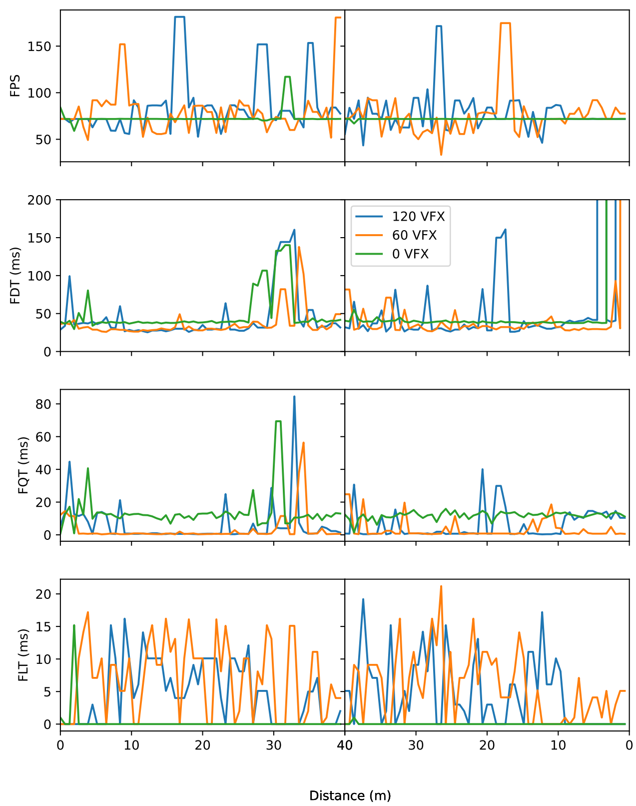

Fig. 5: Behaviour of the FPS (5a), Frame Delivery time (FDT) (5b), Frame Queue Time (FQT) (5c) and Frame Latch Time (FLT) (5d) in milliseconds for the remote rendering experiment with NVIDIA CloudXR.

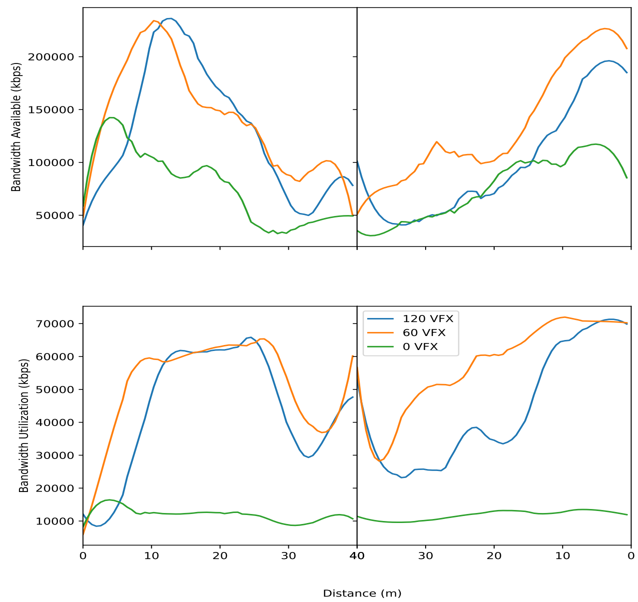

Fig. 6: Behaviour of the Network Bandwidth Available (6a) and Utilization (6b) in kbps for the remote rendering experiment with NVIDIA CloudXR.

utilization without requiring also further bandwidth from the real value available, the other two had a similar trend. An observation we made, is that at the very beginning of the test the bandwidth available start to reach its maximum value. The same happens in the reverse path, from the end to the starting point. We argue that the reason lies in the fact that it is used a bandwidth adaptation algorithm which increases gradually the bandwidth available.

3) Delay (Figure 7): Whilst Bandwidth available was decreasing going far, on the contrary, the RTD was increasing, but the trend is similar in all the three cases. In particular at 40 m, the timestamp in which the user has reached the end of the path, in all the three experiments we obtain the maximum delay which is in the range of 30-50 ms.

C. Remote Rendering with Oculus AirLink

In this last experiment, we focused only on the Motion-to-Photon latency parameter (Figure 8), since it was the most important value we can measure with the Oculus Debug Tool, related to our research. From the graph, we can see that in the first and second cases, we reached values completely unideal for a good experience while walking far from the router, exceeding the recommended 20 ms, synonymous of probable VIMS. On the contrary, the third experiment obtained optimum results almost all the time. We thought that in the case of remote rendering, the MTP latency parameter is no more linked to the power of the graphic card of the HMD, instead, it is related to network quality and bandwidth available. More is the Bandwidth available, more faster the HMD will send head movement to the server and reduce the possibility of VIMS. To resume the results obtained, we can compare local and remote rendering, and in particular the FPS values registered in the three experiments. From Table II, we can see that, whilst with low demanding graphics, both rendering are optimal for the game, the same does not hold for the worst-case scenario, in which remote rendering is obtained

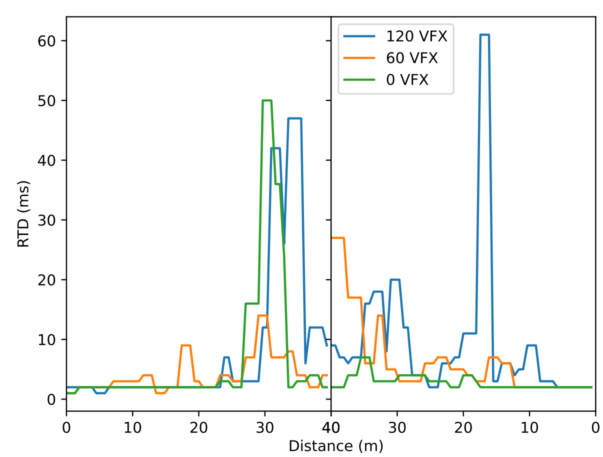

Fig. 7: Behaviour of the Network Round Trip Delay in ms for the remote rendering experiment with NVIDIA CloudXR.

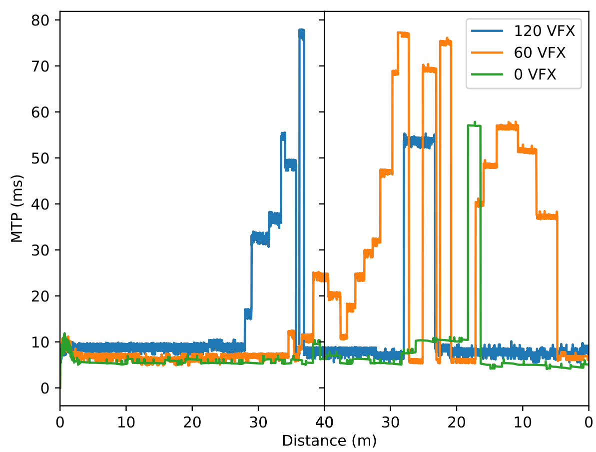

Fig. 8: Behaviour of the Motion-to-Photon latency (MTP) in milliseconds for the remote rendering experiment with Oculus AirLink.

TABLE II: Comparison of FPS, minimum, maximum and average, obtained in local and remote rendering with NVIDIA CloudXR.

Parameter	Quality Settings	Value	Local Rendering	Remote Rendering
FPS	Low Demanding	min	64.61	59.00
		max	72.00	117.10
		avg	70.03	72.4895
	Medium Demanding	min	48.55	33.30
		max	59.82	180.70
		avg	55.89	78.75
	High Demanding	min	28.40	43.40
		max	51.76	181.40
		avg	41.63	82.12

on average two times the value obtained by local rendering. The reason for this is that if local rendering is preferring a better image quality against a higher frame rate, instead remote rendering, with the use of video-compression, tries to reduce image quality for a more stable and higher frame rate. As a side note, we claim that, since all the remote experiments were conducted with the entry-level NVIDIA graphic-card RTX 3060, the same tests would have received a higher FPS value with high-end GPU, like NVIDIA RTX 3080. In fact, the cloud-gaming service of NVIDIA, GeForce NOW[10], has different plans that can be purchased depending on the GPU used in the remote configuration.

From Table III we can also notice that the network parameter RTD is also influenced by the graphics quality. In fact, the worst-case has on average the highest delay among the three experiments. We claim that this is also due to the more bandwidth usage. On the contrary, the average QoE tend to decrease.

V. Conclusions and Future Works

In this paper we tried to find the bottlenecks of VR Cloud-gaming, stressing graphics quality along with the distance from the router, in the worst-case scenario of low radio signal

[10]https://www.nvidia.com/en-us/geforce-now/

TABLE III: Comparison of Bandwidth, Round Trip Delay and Quality of Experience minimum, maximum and average value with the three quality settings in the case of remote rendering with NVIDIA CloudXR.

Parameter	Quality Settings	Value		
		min	max	avg
Bandwidth Available (Mbps)	Low	32.67	142.36	76.69
	Medium	48.19	234.10	137.82
	High	40.69	236.17	114.10
Bandwidth Utilization (Mbps)	Low	8.26	16.43	11.88
	Medium	5.96	71.97	54.60
	High	8.42	71.38	44.11
Round Trip Delay (ms)	Low	1.00	5.00	6.82
	Medium	1.00	14.00	4.13
	High	1.00	47.00	8.60
Quality of Experience	Low	0	5	3.69
	Medium	0	5	3.57
	High	0	5	3.32

and reduced bandwidth available, comparing some of these results with the case of classic VR-Gaming, finding situations in which Remote Rendering have better performances than Local Rendering.

The parameters taken into consideration were FPS, Bandwidth, Delay and QoE. From the experiments we have conducted, Remote rendering obtained better performances from the point of view of FPS. Indeed, the worst-case scenario was managed better than Local Rendering. Furthermore, also with the lowest bandwidth available, FPS did not go down drastically, but remained stable in a little range, thanks to the video compression algorithm implemented by NVIDIA CloudXR. Another parameter affected by the network was the Motion-to-Photon latency, measured in the case of Remote Rendering with Oculus AirLink. In particular, the parameter was increasing linearly with the distance from the router, exceeding also the value of 20 ms, optimal to avoid Visually Induced Motion Sickness.

Future works will be related to trying the same test with NVIDIA CloudXR deployed on a remote server. This will add in general some delay, due to the distance between client and server. One trial will be also to measure battery performance and do a comparison in both the cases of local and remote rendering.

References

[1] S. T. Kumar, "Study of retail applications with virtual and augmented reality technologies," *Journal of Innovative Image Processing*, 2021.

[2] X. Fan, Z. Chai, N. Deng, and X. Dong, "Adoption of augmented reality in online retailing and consumers' product attitude: A cognitive perspective," *Journal of Retailing and Consumer Services*, vol. 53, p. 101986, 2020.

[3] S. S. Alam, S. Susmit, C.-Y. Lin, M. Masukujjaman, and Y.-H. Ho, "Factors affecting augmented reality adoption in the retail industry," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 7, p. 142, 2021.

[4] O. Kutter, A. Aichert, C. Bichlmeier, J. Traub, S. Heining, B. Ockert, E. Euler, and N. Navab, "Real-time volume rendering for high quality visualization in augmented reality," pp. 104–113, 2008.
[5] M. C. de Farias Macedo, A. L. A. Júnior, A. C. dos Santos Souza, and G. A. Giraldi, "High-quality on-patient medical data visualization in a markerless augmented reality environment," *SBC Journal on Interactive Systems*, vol. 5, pp. 41–52, 2014.
[6] S.-G. Shin, J. Yeom, Y. Son, and K.-S. Choi, "Implementation of real-time computer-generated integral imaging system for head mounted display using opengl shading language," *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 232–236, 2019.
[7] S. S., "Virtual reality gaming technology for mental stimulation and therapy," *Journal of Information Technology and Digital World*, vol. 01, pp. 19–26, 09 2019.
[8] A. Lindblom, T. H. Laine, and H. S. Rossi, "Investigating network performance of a multi-user virtual reality environment for mining education," *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–8, 2021.
[9] S. Kavanagh, A. Luxton-Reilly, B. Wuensche, and B. Plimmer, "A systematic review of virtual reality in education," *Themes in Science and Technology Education*, vol. 10, no. 2, pp. 85–119, 2017.
[10] J. Bos, "Visual image safety," *Displays*, no. 32, pp. 151–152, 2011.
[11] C. Diels and P. Howarth, "Frequency characteristics of visually induced motion sickness," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 55, no. 3, pp. 595–604, 2012.
[12] K.-L. Chan, K. Ichikawa, Y. Watashiba, and H. Iida, "Cloud-based vr gaming: Our vision on improving the accessibility of vr gaming," *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*, p. 1, 2017.
[13] S. S. Sabet, S. Schmidt, S. Zadtootaghaj, B. Naderi, C. Griwodz, and S. Möller, "A latency compensation technique based on game characteristics to mitigate the influence of delay on cloud gaming quality of experience," *MMSys '20: 11th ACM Multimedia Systems Conference*, pp. 15–25, 2020.
[14] U. Lampe, Q. Wu, S. Dargutev, R. Hans, A. Miede, and R. Steinmetz, "Assessing latency in cloud gaming the accessibility of vr gaming," *Communications in Computer and Information Science (CCIS)*, pp. 1–17, 2014.
[15] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei, "Measuring the latency of cloud gaming systems," *Association for Computing Machinery*, p. 1269–1272, 2011. [Online]. Available: https://doi.org/10.1145/2072298.2071991
[16] A. D. Domenico, G. Perna, M. Trevisan, L. Vassio, and D. Giordano, "A network analysis on cloud gaming: Stadia, geforce now and psnow," *Network*, pp. 247–260, 2021.
[17] T. Kushida and Y. Shibata, "Framework of end-to-end performance measurement and analysis system for internet applications," *International Conference on Information Networking*, pp. 674–679, 2001.
[18] A. A. Simiscuka and G. Muntean, "Synchronisation between real and virtual-world devices in a vr-iot environment," *Proc. of the IEEE International Symposium on Broadband Multimedia Systems*, pp. 1–6, 2018.
[19] V. Parthasarathy, A. A. Simiscukay, N. O'Connorz, and G.-M. Muntean, "Performance evaluation of a multi-user virtual reality platform," *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1–6, 2020.
[20] J. Zhao, R. Allison, M. Vinnikov, and S. Jennings, "Estimating the motion-to-photon latency in head mounted displays," *2017 IEEE Virtual Reality (VR)*, pp. 313–314, 03 2017.
[21] M. Liubogoshchev, K. Ragimova, A. Lyakhov, S. Tang, and E. Khorov, "Adaptive cloud-based extended reality: Modeling and optimization," *IEEE Access*, vol. 9, pp. 35 287–35 299, 2021.
[22] X. Zhou, C. Yao, C. Fukai, L. Kan, L. T. Tian, X. Rufeng, and L. Liquan, "Remembering urban village: Using cloudxr technology as an enhanced alternative to better disseminate heritage," *2022 POST-CARBON - Proceedings of the 27th CAADRIA Conference*, pp. 757–766, 2022.
[23] Y.-C. Li, C.-H. Hsu, Y.-C. Lin, and C.-H. Hsu, "Performance measurements on a cloud vr gaming platform," *QoEVMA'20: Proceedings of the 1st Workshop on Quality of Experience (QoE) in Visual Multimedia*, pp. 37–45, 2020.