A Study on Energy Efficiency in Edge-assisted VR Applications with Meta Quest 2 for Disaster Management

Lorenzo Romagnoli, Gabriele Proietti Mattia, Roberto Beraldi
Department of Computer, Control and Management Engineering "Antonio Ruberti"
Sapienza University of Rome
Rome, Italy
ORCID: 0009-0004-2008-7841, 0000-0003-4551-7567, 0000-0002-9731-6321

***Abstract*—Virtual reality (VR) is a technology that has the potential to innovate whole sectors as well as the way we interact with digital environments. AR and VR devices are a perfect way to gain knowledge, information, and to practice operation plans during, before, and after a disaster took place, exploiting their capabilities. VR headsets like the Meta Quest 2 allow to perform rendering of applications locally or exploit the offloading to a Server in the cloud, and, in a disaster management scenario, the energy consumption of the device is a fundamental knowledge in order to make rational choices on which of the two types of rendering to perform. In this paper, we investigate the energy efficiency of the Meta Quest 2 in both of the two rendering approaches performing tests based on a benchmark application developed in Unreal Engine. From the results of our experiments, we found that remote rendering, performed via AirLink, allowed us to obtain higher FPS and overall quality, as well as keep the device GPU and CPU usage at lower values than with the local rendering paradigm. However, from the energy efficiency perspective, test results showed that besides the lighter CPU and GPU work using the remote rendering approach, the energy consumption, in the overall execution, exceeds the one using the local rendering paradigm, because of the network communication with the server.**

***Index Terms*—VR, remote rendering, Edge Computing, Meta Quest 2**

I. INTRODUCTION

Virtual reality headsets are one of the technical breakthroughs that have gained the public's interest the most. These innovative technologies have changed the way we interact with digital environments, blurring the perceived boundary between real and virtual worlds. Beyond entertainment, these devices have opened up previously unimaginable possibilities for education [1] [2], training, and even therapeutic experiences thanks to their immersive characteristics and the ability to replicate actual environments. In fact, they provide a safe and engaging environment for learning, and growing skills and knowledge in a wide range of subjects and situations, from dentistry [3] to engineering. VR headsets have become an indispensable tool in many industries thanks to their revolutionary potential, which has redesigned fields that go from architecture to healthcare [4] and tourism. The advance in the virtual and augmented reality industry led to the implementation of innovative features that could and have found application in almost every field. We can cite the Passthrough feature[1] of Meta Quest 2[2] as one of them. This mixed reality feature allows users to see the outside real world while wearing the headset, taking advantage of the cameras placed on the device that are then able to show, at the same time, information and data on the displays, based on the specific underlying application and user needs.

In this paper, we are studying the energy efficiency of the Meta Quest 2 in order to understand and compare the energy constraints that are related to the device during edge-assisted and standalone usage. Those constraints would be fundamental knowledge for the purpose of using this kind of device in disaster management scenarios such as an earthquake or a flood, just to cite some of the areas or situations of application. In fact, in the field of disaster management, virtual reality has demonstrated outstanding potential that could be exploited by emergency responders, companies, and communities. VR devices can be a precious resource in better understanding, preparing for, and managing many sorts of catastrophes by offering realistic and immersive simulations that can be adapted to actual and specific needs. We can think about the usage of VR headsets as a perfect way to gain knowledge about buildings or city areas after a disaster happened and to allow the rescue teams to get precious information about the disaster site in an immersive way. Being able to perform simulations or training based on the plan being studied for a disaster recovery situation [5] [6] [7], disaster prevention and its related education [8] [9], these are only some of the possibilities and opportunities offered by this type of devices. Mixed reality features then could be exploited in order to provide precious information and coordinate emergency responders directly on site. Those features could allow them to have visually instant information on the display of the device in order to have more clear and complete operative knowledge while being able to keep the focus on the surrounding real environment.

The rest of this paper is organized as follows. Related works are presented in section II. In section III we define the

---

[1] https://developer.oculus.com/blog/mixed-reality-with-passthrough

[2] https://www.meta.com/it/quest/products/quest-2

experimental setup and the test settings. Test outcomes are presented in section IV. Section V ends the paper with a resume of the study results and future works.

II. RELATED WORK

Several studies focused on edge nodes and the related computation offloading [10] [11]. In particular, the exploit of edge nodes in massively multiplayer online VR games has been discussed by Zhang et al. [12] which focused on latency, bandwidth, and offloading between devices, edge nodes, and center clouds in this particular scenario that involves a high number of players of the game. The study of energy efficiency on VR devices was also addressed by Leng et al. [13] which focused on energy consumption in the case of video processing. In their study, they propose an end-to-end system capable to allow energy savings in this kind of device up to 42% focusing on the costly projective transformations performed on the device. Du et al. [14] addressed the task of 360° video rendering on VR devices exploiting multi-access edge computing, terahertz wireless networks, and a deep reinforcement learning approach for the offloading with a view on latency, bandwidth, and energy efficiency. Studies with the Meta Quest 2 were conducted by Maiorano et al. [15] with a focus on VR headsets trade-offs in quality of experience based on bandwidth, connection quality, and speed. The studies were conducted on the same device used for the tests performed for this paper, to understand trade-offs and differences related to QoE using the local and remote rendering paradigms, the last one taking into consideration an offloading of the rendering computation both on a cloud server and an edge node situated in the same network of the VR device. Lin et al. [16] studied how Pervasive Edge Computing (PEC) could be a promising method for wireless VR experiences. Their research is based on the idea of performing an offloading of the viewport rendering in which the resource allocation problem is transformed into a Markov Decision Process while an RL-based online learning algorithm is used to define the optimal policy. Quantum parallelism is then integrated into the RL to improve learning efficiency. The study is based on the fact that viewport rendering is a computation-intensive task in which the VR headset has to constantly perform the rendering while exchanging data over the network. The study addresses not only the quality of experience (QoE) but also the energy efficiency of VR devices taking into consideration not only the rendering complexity but also the related data transmission.

However, none of the cited works focused on the study of the energy performance of the Meta Quest 2, proposing a comparison between local and remote rendering based on application testing. Moreover, some of the cited works focused on the computational complexity of the rendering, taking as a parameter for the studies the intensity of the work on the CPU and GPU of the VR device, without considering the whole energy consumption of the headset during its usage. As we will show in our experimental results, in fact, the data interchange over the network, with the server, is a source of power consumption that should be taken into account to understand the energy efficiency of the device.

III. PROPOSED METHOD

A. Experimental setup

In our experiments, we used a Meta Quest 2, 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. The device is equipped with two touch controllers with accelerometers and gyroscopes, and six degrees of freedom (6DOF) inside-out tracking through 4 built-in cameras.

As a server, we used a PC with 32GB RAM, Intel Core i7-6700K CPU, and an NVIDIA GeForce GTX 1080 GPU, connected to the router via Gigabit Ethernet.

To perform the tests a benchmark application has been developed through the engine Unreal Engine (UE) at version 5.1.1[3]. The application is developed such that the camera automatically follows a fixed path without the possibility for the user to move its view neither moving the headset nor using the touch controllers in order to provide the replicability of the tests. Following the fixed path all benchmark execution tests have a related running time of 129s. The main level of the application is composed of 4 main interconnected rooms of increasing rendering complexity after which we find a final fifth low-complexity room which is the one in which the application automatically closes itself to end the test. In Figure 1 is possible to observe the room composition of the main level of the benchmark application while in Table I is possible to observe the data related to their complexity. Regarding the profiling of the performance metrics (see Table II), during the execution of the tests described in the next subsections the VrApi Logcat logs[4] are used in order to retrieve information about CPU and GPU usage and frequencies other than the FPS values for all tests scenario except the one regarding the execution of our benchmark application via remote rendering. Concerning the profiling of energy consumption during the tests, it has been developed and installed on the device an android application that starts a service running in the background with the purpose of retrieving and saving in a file, battery-related data during the execution of the tests.

Room	Meshes count	Triangles count	Lights count	VFX count
Room 1	92	585.827	0	0
Room 2	134	1.572.651	3	0
Room 3	258	3.454.169	21	0
Room 4	70	223.285	0	78
Room 5	47	564	0	0

TABLE I
COMPOSITION AND COMPLEXITY OF THE MAIN LEVEL OF THE APPLICATION.

---

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

[4] https://developer.oculus.com/documentation/native/android/ts-logcat

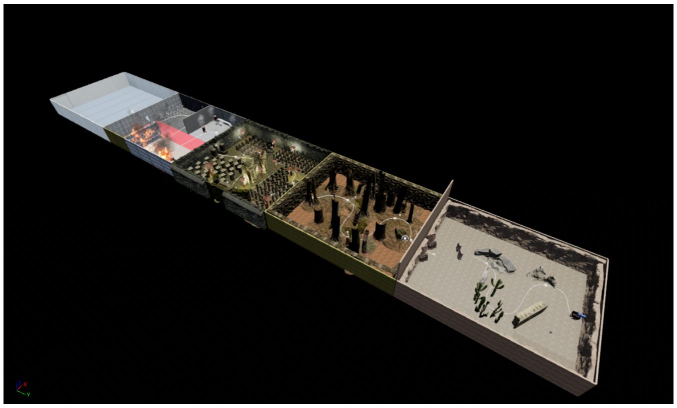

Fig. 1. Main level of the benchmark application.

B. Idle Test: Initial VR headset assessment

1) Home Environment Test: The first test setup involves the usage of the VR headset's Home Environment as a benchmark with the same test duration as the developed Unreal Engine application in order to be able to compare the results. This landing environment requires a constant rendering from the device since it is not a static image but instead, an interactive environment in which the user can move around, therefore requiring similar rendering computations of any other VR game or application. The cited environment is used to retrieve baseline data to compare and evaluate the further tests performed. We decided to use this Home Environment since the rendering complexity is lower than the Unreal Engine application developed by us for the tests, allowing us to understand the battery consumption in different rendering complexity scenarios.

2) Home Environment with network Usage Test: In this second test setup, we involve the same rendering scenario of the Home Environment to which we added heavy usage of the network. To perform this task we developed an Android application with the purpose to perform the download of a high-size file of 10GB stored on the server PC, performing this download using our local network in order to maximize the network bandwidth usage. This test was designed in order to implement an intensive network usage similar to a remote rendering scenario of execution of an application.

In both of the Home Environment tests, the experiments were conducted leaving the VR headset in the same position once the tests started in order to make the device render the same portion of the environment for the whole test duration.

C. Local rendering

In the case of the local rendering test setup, the developed benchmark application is installed directly on the device so that the rendering pipeline is fully managed by it.

D. Remote rendering

In the case of remote rendering, we used the AirLink[5] feature in order to perform the rendering on our PC server (see Figure 2). Using this approach the rendering pipeline is managed by the server that receives inputs from the device and sends back the rendered frames, then the VR device decodes the frames rendered on the server and displays them. Furthermore is important to mention that the AirLink feature used in our tests is developed to work only in a LAN setting. For this test setup, the FPS values are retrieved using Unreal Engine commands. Specifically, we used the command "Start/Stop FPSChart", from which we retrieved the time needed (in seconds) to render a single frame, which we call d_r. Given those data, we calculated the frame rate (FPS) as d_r^{-1}.

After obtaining those results we were able to compute the average of those FPS values for each second of the benchmark execution time and present the final averaged values for the FPS.

Both the local and remote rendering tests have been executed using the same application, given the capability of Unreal Engine to build the application for both Android and Windows systems, starting from the same UE project.

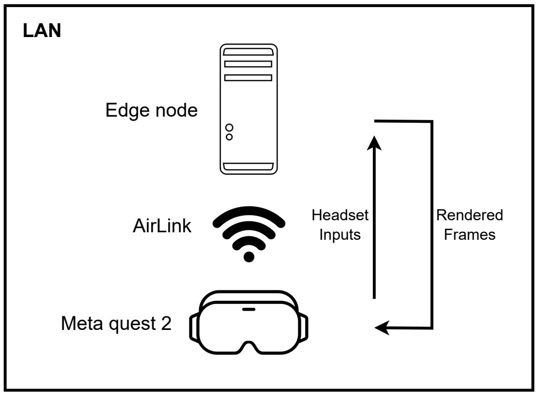

Fig. 2. Architecture of the LAN settings for the remote rendering via AirLink.

IV. EXPERIMENTAL RESULTS

In this section, we present the results that we obtained firstly during the Home Environment testing (Section III-B1) which served to derive baseline values of CPU, GPU, and energy consumption of the headset. After that, we present the results obtained during Home Environment testing with network usage (Section III-B2). Then we present the results obtained during the Local (Section III-C) and Remote (Section III-D) rendering tests. Finally, in Section IV-C we summarize all the results and apply statistics to confirm the findings.

The results presented in this section are given from the execution of 10 tests for each of the test setups, from them the metrics, so the recorded values have been averaged between

---

[5] https://www.meta.com/it-it/help/quest/articles/headsets-and-accessories/oculus-link/connect-with-air-link/

the tests (for each test scenario) in order to obtain the average value per second for each metric during the test execution time. The performance metrics considered during the tests are defined in Table II.

Name	**Meaning**
FPS	Number of frames per second rendered
CPU frequency	The clock speed of the CPU
GPU frequency	The clock speed of the GPU
CPU usage	CPU utilization percentage
GPU usage	GPU utilization percentage
Battery voltage	The battery voltage in millivolts
Battery current	The current coming from the battery in milliamps
Battery wattage	Instant power absorption of the device in milliwatt

TABLE II
PARAMETERS TAKEN INTO CONSIDERATION DURING THE TESTS.

A. *Idle Test*

1) Home Environment Test: In Figure 3 is possible to observe the charts related to the metrics regarding the CPU and GPU data. The obtained CPU frequency during the test was stable at 1171MHz while the GPU was working at the maximum clock speed of 525MHz. With respect to the usage, instead, we got an average of 24% for the CPU while the GPU was at an average of 81%. With respect to battery-related data, once we retrieved the values related to the Voltage and Current flowing from the battery, we proceeded to calculate the instant power consumption where we found an average value of 5668mW.

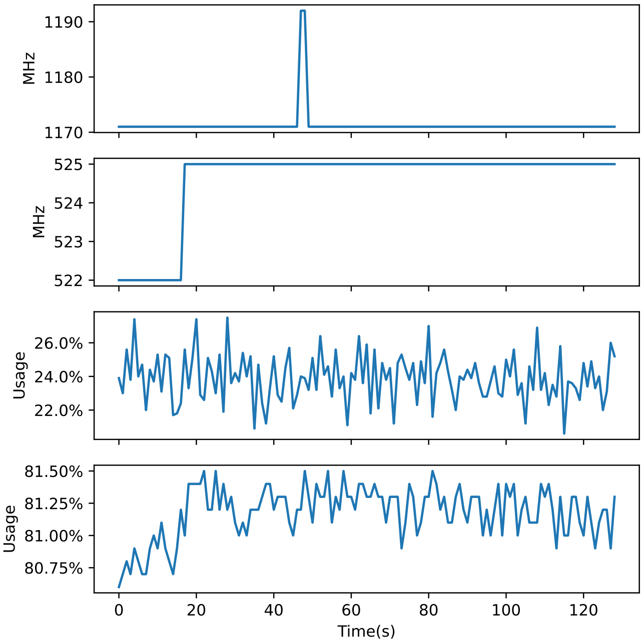

Fig. 3. Behavior of the CPU frequency (3a), GPU frequency (3b), CPU usage (3c), GPU usage (3d) during the Home Environment test.

2) Home Environment with network Usage Test: In this second test, we noticed the same frequencies as the previous, average usages instead increased to 35% and 87% for the CPU and GPU respectively. The increase was due also to the different orientation of the headset during the execution of the test which made the device render more complex parts of the Home Environment. In fact, in the Home Environment, the headset's tracking is used to provide the immersive experience of VR without the possibility to fix that constraint. Finally, the battery-related data showed the battery-draining behavior in the presence of network utilization. From Figure 4 is possible to note the difference in power consumption between the two tests, with an average of 5668mW for the first one and an average of 6787mW for the second one in which the network usage is involved. FPS were stable at 90FPS for the whole time in both the Home Environment tests.

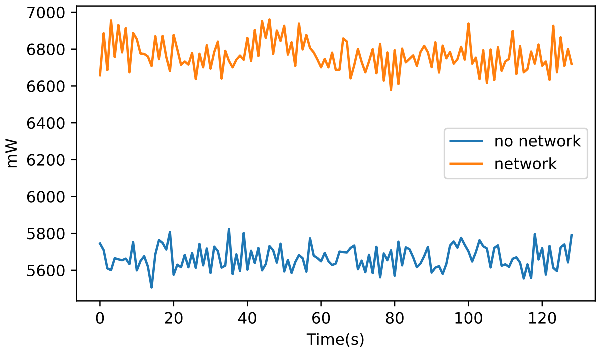

Fig. 4. Trace of the power absorption during the Home Environment tests.

B. *Local and Remote Rendering*

In the following charts, we used vertical green lines to indicate the seconds during the test execution time in which there is a shift from one application room to the other one (see Table I).

1) Frame rate: Figure 5 shows the FPS as a function of time for the rooms defined in Table I. The FPS observed under local rendering was lower than the nominal value of 72 FPS, required to achieve an enjoyable VR experience and avoid the side effects of virtual reality experiences, also known as Visually Induced Motion Sickness (VIMS), because of the increased rendering complexity. Differently with the remote rendering was able to reach the requested 72 FPS for most of the time except during the rendering of the fourth room.

In relation to the FPS values recorded is important to point out that Unreal Engine automatically detects the hardware of the device on which the application is being executed, tuning the quality settings for the rendering accordingly. The quality of the benchmark application executed on the Meta Quest 2 as a standalone is set to medium quality (level 1), while for remote rendering the quality was automatically set to the highest available (level 3). The drop of frames per second noticeable around the 80th second in the remote rendering scenario is due to the performed rendering of the fourth room, in which the visual effects (VFX=78), computed at maximum quality, increase the rendering complexity above the server hardware capabilities. Similarly, we can notice a drop of frame in the same time region on the local rendering scenario, but thanks to the lower quality preset, the VR headset is able to maintain a slightly higher FPS value inside the mentioned room of the application.

Concerning the local rendering scenario, from the figure is possible to notice some spikes in the value recordings, e.g. three spikes in room 1; these are related to the path followed by the camera through which the user is looking during the execution of the benchmark. When the camera points to the side walls of each room, the amount of objects and details captured by the view and so to be rendered is lower since just a fraction of the overall objects of the room are in the view of the user, leading to an increase of the FPS that the device is able to reach for a small number of seconds.

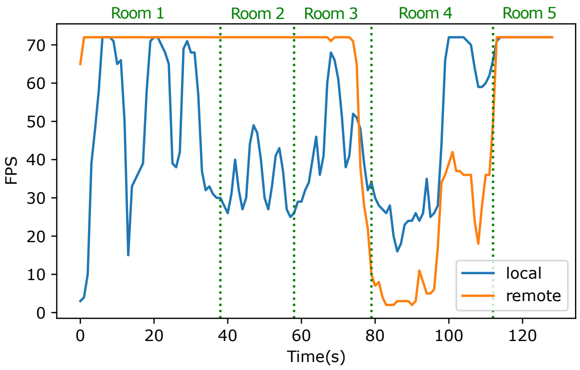

Fig. 5. Behavior of the FPS for different rooms during the local and remote tests.

2) CPU and GPU frequency: With respect to the frequencies to which CPU and GPU (see Figure 6) are running during the execution of the tests, it is possible to observe how the clock speed of the GPU is kept at the maximum speed of 525MHz during the two types of execution where is possible to observe a stable frequency during the whole running time. As far as the CPU is concerned, the remote rendering keeps the frequency of the processor stable at 1382MHz while the local rendering shows spikes related to the FPS, meaning that when the rendering complexity decreases the CPU is less stressed.

3) CPU and GPU usage: Figure 7 shows CPU and GPU usage. We found the utilization is lower and more stable during remote rendering. In fact, remote rendering only requires decoding the frames that are sent via the network and displaying them to the user, which are light operations that load the CPU for less than 20 % of the time.

Local rendering, as expected, required a higher usage of the CPU and GPU, with the latter showing an average usage of 87%, due to the whole rendering pipeline managed by the device and the rendering complexity of the application developed. Moreover, is possible to notice how the usage of the CPU is strictly related to the FPS at which the device is able to render the application (see Figure 5). We can note how an increase in FPS corresponds to an increase in CPU usage, this is due to the fact that the processor has the task of managing the frames rendered by the GPU to perform the display of them.

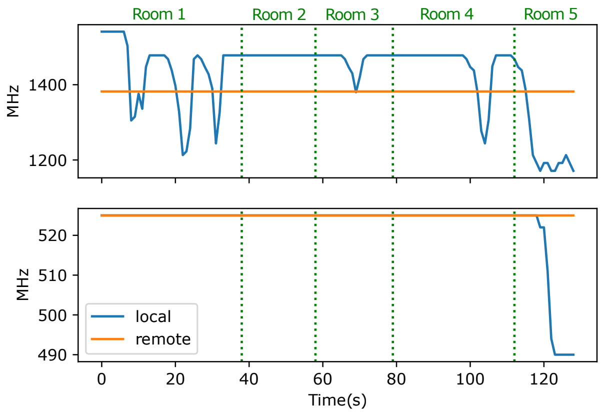

Fig. 6. Behavior of the CPU frequency (top), GPU frequency (bottom) during the local and remote tests.

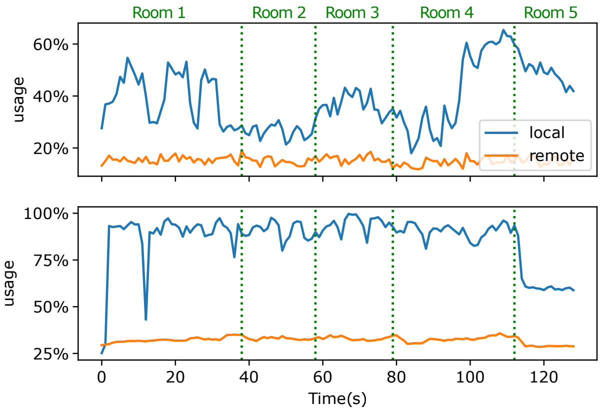

Fig. 7. Behavior of the CPU usage (top), GPU usage (bottom) during the local and remote tests.

4) Energy consumption: With respect to battery-related data, once we retrieved the values related to the Voltage and Current flowing from the battery, we proceeded to calculate the instant power consumption in order to better understand the battery draining behavior as follows:

$$mW = A \times mV$$

From the results, we had the confirmation of higher energy consumption using the remote rendering paradigm than the local one. In fact, we obtained an average value of 6525mW for the local rendering that goes up to 6862mW in the remote rendering scenario. From the chart in Figure 8, especially using the local rendering paradigm, we can note how the trace of power absorption is correlated to the complexity of the various rooms of the application; the more the rendering complexity of a room, the more is the battery draining noticed.

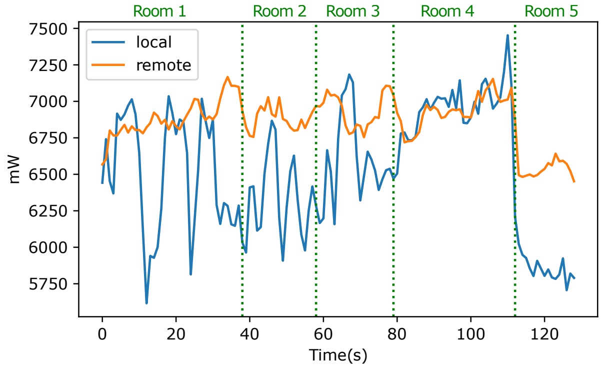

Fig. 8. Trace of the power absorption during the local and remote tests.

C. F- and t-test Statistics

We finally summarize all the results obtained in the proposed experiments. In Tables III and IV, where we present the average instant power consumption values during the tests and the reference of CPU and GPU usage, we labeled as “Idle” the results of the Home Environment test (Section III-B1) while “Idle network” the results of the Home Environment test with network (Section III-B2). Then, “Local” (Section III-C) and “Remote” (Section III-D) labels instead indicate the tests conducted using our developed benchmark application executed using the two different rendering paradigms.

We proceeded to conduct a Student’s t-test in order to confirm that there was a significant difference between the average instant power consumption values recorded in all the different test scenarios. However, we previously conducted the F-test to check if there was a difference between the variances among the tests to drive the t-test correctly. The F- and t-tests have been conducted between each possible pair of test scenarios with a confidence value equal to 95%. The F-tests results denoted different variances in all test pairs taken into consideration but one that is the pair of tests related to the Home Environment with network usage and the one using our benchmark application via the remote rendering paradigm (the two are denoted with a “*” in Table III). Given this pair of tests, in fact, we obtained from the F-test a resulting equal variance. The t-test results then confirmed that between each pair of tests, there is a significative difference in their instant power consumption average values.

V. Conclusions and Future Works

In this paper, we focused on the energy efficiency of the Meta Quest 2 VR headset in order to understand its capabilities and limitations in a disaster management scenario with edge-assisted application execution. The performance parameters taken into consideration were FPS, CPU and GPU frequencies and usage, battery voltage, current, and instant power consumption. The tests denoted how edge-assisted applications are more demanding from an energy point of view than locally rendered applications. The metrics readings indeed, showed that even a lighter amount of CPU and GPU usage on the device could bring to a faster battery draining in case of utilization of the network connection. Furthermore, comparing the instant power consumption of the remote rendering and the Home Environment with network usage tests, denoted how the data interchange performed during an edge-assisted application execution is costly from a power consumption perspective. These findings are useful, in the end, to rationally decide, based on the specific needs, which paradigm to adopt in the use of the device taking into consideration its battery lifetime. Depending on the scenario, it could be possible to build applications running on the headset as a standalone, which will allow it to use less possible energy. We can think of pre-built applications for training, information gathering, or briefing to prepare operations after a disaster. In case sending real-time information over the network is fundamental, given the specific scenario, the operation could be carried out by exploiting the headset capabilities, but keeping in mind the more energy consumed by the device due to the network usage.

	Remote	**Local**	**Idle**	**Idle net.**
Test 1	6878	6241	5632	6755
Test 2	6905	6217	5657	6789
Test 3	6998	6242	5672	6875
Test 4	6976	6349	5599	6880
Test 5	6959	6726	5664	6653
Test 6	6773	6586	5693	6703
Test 7	6771	6695	5646	6994
Test 8	6804	6731	5713	6625
Test 9	6774	6754	5704	6656
Test 10	6783	6711	5704	6752
Total average	6862	6525	5668	6768
Variance	8512*	54257	1329	14033*

TABLE III
AVERAGE INSTANT POWER CONSUMPTION (MW) OF META QUEST 2 IN THE PROPOSED TEST SCENARIOS.

	Remote	**Local**	**Idle**	**Idle net.**
CPU usage	15%	39%	24%	35%
GPU usage	32%	87%	81%	87%

TABLE IV
AVERAGE GPU AND CPU USAGE VALUES OF META QUEST 2 IN THE PROPOSED TEST SCENARIOS.

Future works will be related to defining a function able to describe the impact of the various hardware units of the device on the resulting power absorption. Furthermore, the studies could be directed into implementing an algorithm that, based on specific parameters such as the instant power consumption, will allow performing a switch at runtime between the local and remote rendering paradigm with a view to maximizing the battery lifetime of the device.

References

[1] M. Holly, J. Pirker, S. Resch, S. Brettschuh, and C. Gütl, “Designing vr experiences – expectations for teaching and learning in vr,” *Educational Technology & Society*, vol. 24, no. 2, pp. 107–119, 2021. [Online]. Available: https://www.jstor.org/stable/27004935

[2] Z. Pan, A. D. Cheok, H. Yang, J. Zhu, and J. Shi, “Virtual reality and mixed reality for virtual learning environments,” *Computers & Graphics*, vol. 30, no. 1, pp. 20–28, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849305002025

[3] T.-K. Huang, C.-H. Yang, Y.-H. Hsieh, J.-C. Wang, and C.-C. Hung, “Augmented reality (ar) and virtual reality (vr) applied in dentistry,” *The Kaohsiung Journal of Medical Sciences*, vol. 34, no. 4, pp. 243–248, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1016/j.kjms.2018.01.009

[4] N. Rohrbach, P. Gulde, A. R. Armstrong, L. Hartig, A. Abdelrazeq, S. Schröder, J. Neuse, T. Grimmer, J. Diehl-Schmid, and J. Hermsdörfer, “An augmented reality approach for adl support in alzheimer’s disease: a crossover trial,” *Journal of neuroengineering and rehabilitation*, vol. 16, pp. 1–11, 2019.

[5] Y. Sermet and I. Demir, “Flood action vr: A virtual reality framework for disaster awareness and emergency response training,” 08 2018.

[6] E. Hsu, Y. Li, J. Bayram, D. Levinson, S. Yang, and C. Monahan, “State of virtual reality based disaster preparedness and response training,” *PLoS Currents*, no. APR 2013, 2013.

[7] A. Mossel, M. Froeschl, C. Schoenauer, A. Peer, J. Goellner, and H. Kaufmann, “Vronsite: Towards immersive training of first responder squad leaders in untethered virtual reality,” in *2017 IEEE Virtual Reality (VR)*, March 2017, pp. 357–358.

[8] B. WANG, H. OKAWA, U. TOYAMA, and K. KASHIYAMA, “A tsunami mitigation experience system using vr technology for disaster education,” *Japanese Journal of JSCE*, vol. 79, p. n/a, 03 2023.

[9] K. Meesters and B. V. de Walle, “Disaster in my backyard: A serious game introduction to disaster information management,” in *International Conference on Information Systems for Crisis Response and Management*, 2013.

[10] J. Zhang, H. Guo, and J. Liu, “A reinforcement learning based task offloading scheme for vehicular edge computing network,” in *Artificial Intelligence for Communications and Networks*, S. Han, L. Ye, and W. Meng, Eds. Cham: Springer International Publishing, 2019, pp. 438–449.

[11] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, “Edge cloud offloading algorithms: Issues, methods, and perspectives,” *ACM Comput. Surv.*, vol. 52, no. 1, feb 2019. [Online]. Available: https://doi.org/10.1145/3284387

[12] W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, “Towards efficient edge cloud augmentation for virtual reality mmogs,” *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017.

[13] Y. Leng, C.-C. Chen, Q. Sun, J. Huang, and Y. Zhu, “Energy-efficient video processing for virtual reality,” in *Proceedings of the 46th International Symposium on Computer Architecture*, ser. ISCA ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 91–103. [Online]. Available: https://doi.org/10.1145/3307650.3322264

[14] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, “Mec-assisted immersive vr video streaming over terahertz wireless networks: A deep reinforcement learning approach,” *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517–9529, 2020.

[15] G. Maiorano, G. P. Mattia, and R. Beraldi, “Local and remote fog based trade-offs for qoe in vr applications by using cloudxr and oculu air link,” in *2022 International Conference on Edge Computing and Applications (ICECAA)*, Oct 2022, pp. 95–101.

[16] P. Lin, Q. Song, D. Wang, R. Yu, L. Guo, and V. Leung, “Resource management for pervasive edge computing-assisted wireless vr streaming in industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. PP, pp. 1–1, 02 2021.