Use Case #8

AR-assisted emergency surgical care [CTTC Testbed]

Overview and Objectives

For the first-party experimentation of the 5G-EPICENTRE framework, ORAMA has developed a Unity-based[1] AR application, that facilitates first-aid responders with critical information about an injured patient on the disaster site, using its novel Extended Reality (XR) authoring framework (MAGES) Software Development Kit (SDK). Using an AR headset equipped with this application, first aid responders can have access to step-by-step instructions for a specific surgical task or anatomical information (e.g., organs, vessels, bones), overlayed as deformable objects on top of the patient's body.

A figure showcasing the content of this application is shown below (Figure 1): A first aid responder, equipped with an AR-Head-Mounted Display (HMD), is depicted at a disaster site, where they located a patient. Utilizing the designed application, the responder gains access to step-by-step instructions for critical operations and is able to visualize the patient's veins, bones, and/or internal organs as deformable objects overlaid on the patient's body.

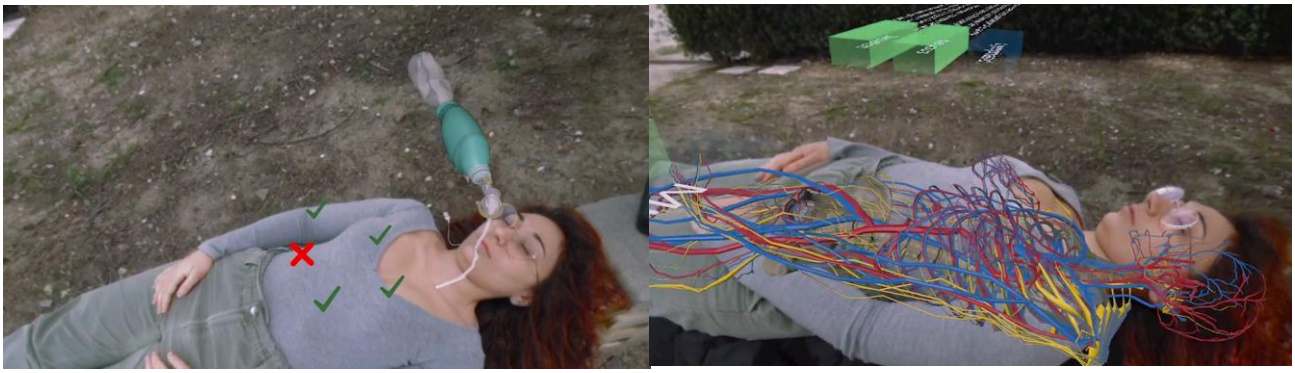

Figure 1 : Overlay on patient's body

The original application was modified to offload the process of scene rendering to a vertical application lying on the edge-cloud continuum, taking advantage of the available Central and Graphics Processing Unit (CPU & GPU) resources to perform the heaviest task in the pipeline, i.e., the scene generation and rendering. The output video stream will be transmitted over the 5G network to the AR headset, where a user-facing application was deployed to receive, decode and project it on the HMD's screen. The latter application is also used to track and send the headset's input (movement and controller triggers) to the edge-residing network application, so that the scene and game play is updated, based on the user's actions.

The objectives of the experimentation were to measure end-to-end (E2E) network latency for video streaming from the edge resources to the HMD, the maximum aggregated bandwidth, and packet loss. Various experiments under different settings were conducted to optimize these metrics. In each experiment, in addition to recording these metrics, we also noted the qualitative result, i.e., the visual-driven QoE of the AR user. Additionally, we measured the HMD's energy consumption by recording its battery drain.

[1] www.unity.com

This project has received funding from the European Union's Horizon 2020 Innovation Action programme under Grant Agreement No 101016521.

Use Case Description

Surgeons should play a central role in disaster planning and management due to the overwhelming number of bodily injuries that are typically involved during most forms of disaster. In fact, various types of surgical procedures are performed by emergency medical teams after sudden-onset disasters, such as soft tissue wounds, orthopaedic traumas, abdominal surgeries [1] [2]. HMD-based Augmented Reality (AR), using state-of-the-art hardware such as the Magic Leap or the Microsoft HoloLens, has long been foreseen as a key enabler for clinicians in surgical use cases [3], especially for procedures performed outside of the operating room. In such condtions, monolithic HMD applications fail to maintain important factors such as user mobility, battery life, and Quality of Experience (QoE), leading to considering a distributed cloud/edge software architecture. Toward this end, 5G and cloud computing will be a central component in accelerating the process of remote rendering computations and image transfers to wearable AR devices.

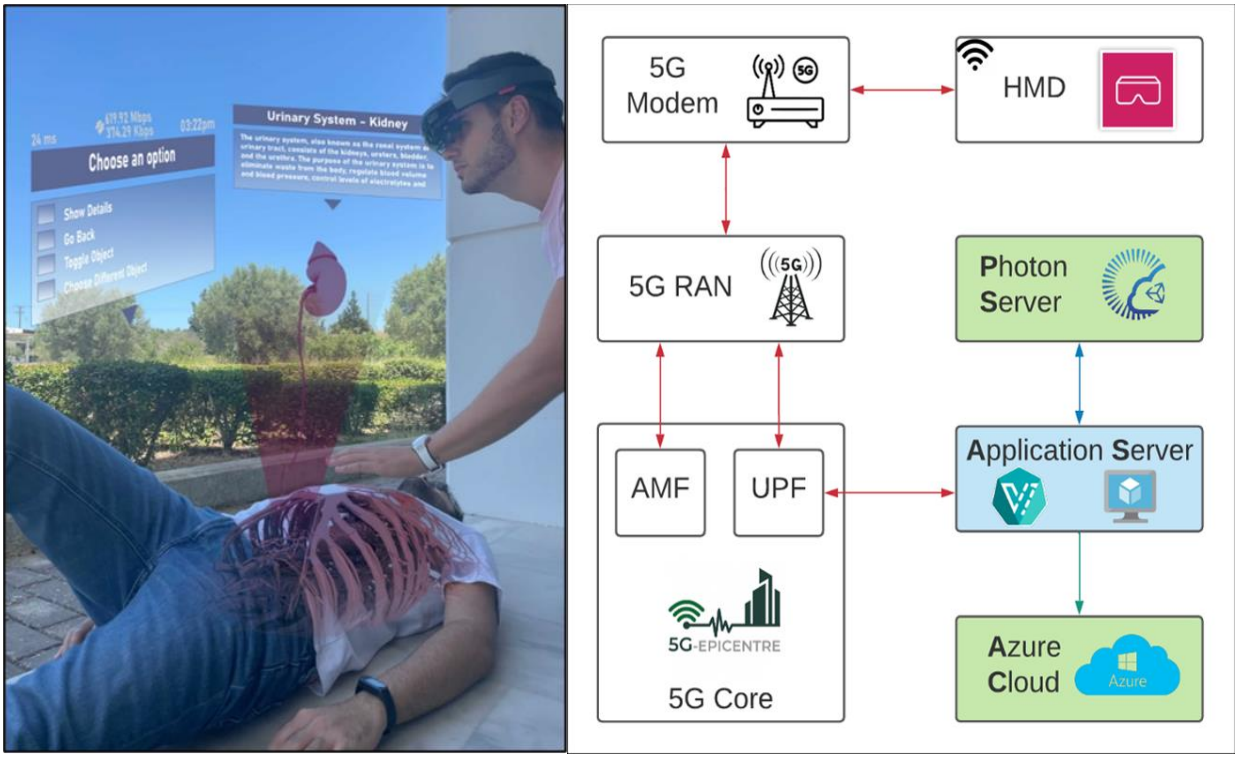

Figure 2: UC8 - The PPDR responder uses an AR HMD to see overlayed info and deformable objects on top of the patient. Envisioned example of UI layout and ayout of the UC components.

Experiment Setup/Methodology/Deployment

The experimentation setup in the CTTC testbed for UC8 is similar to the one in UMA testbed, where, instead of a standalone Windows 11 machine, the UC8 edge server component was deployed via libvirt2 on a LXD container, as a VM running Windows 10. The edge node was equipped with an NVIDIA GeForce RTX 2080 Ti GPU card, suitable for the needs of a high-fidelity AR application. The CPU of the VM is an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz, with 16GB of RAM.

The deployed VM needed exclusive use of the GPU, with GPU pass-through, since Virtual GPU licensing was not available in the testbed. To provide GPU access, the KubeVirt K8 module was used, which can provide hardware devices to libvirt.

After the experimentation was concluded, we also tested a standalone Windows-based machine, similar to the UMA setup. However, this configuration yielded results identical to the original setup, so we have opted not to provide further details. The AR headset used was the Hololens 2.

The deployment involved using Windows-based (Virtual) Machines instead of containers. Although docker containers outperform VMs in the case of space and processing overhead, they are rather immature in graphics acceleration processes. In this case, the use of VMs is far more advantageous, since they have highly optimized graphics drivers and Kernel-based VM (KVM) GPU passthrough support. Furthermore, docker containers have limited VR drivers support, since only experimental versions (for all vendors) for Linux are currently available. Additionally, Unity's support for Linux is still at an experimental stage as well, making the task of porting the ORAMA's MAGES SDK to Linux a difficult and error prone procedure, as VR support for Linux remains a critical open issue.

For all the reasons above, we have opted for using standalone Windows-based (virtual) machines that will allow us to deploy the designed AR application in a manner that will deliver high QoE for the end-user, i.e., the first aid-responder, which is crucial for all AR applications. This setup allows the achievement of KPIs, which indicate its successful leverage to a remote-rendering pipeline, exploiting the low latency and high bandwidth capabilities of the 5G networks. Lastly, this UC demonstrates the capacity of the platform and testbed architectures, to host VM-based vertical systems, besides containerized ones.

[2] libvirt: The virtualization API

Experiment Execution and Results

Below we provide the results of the two types of experiments conducted at the CTTC testbed, focusing on network metrics such as RTT, throughput, and packet loss. Both experiments' settings were selected towards achieving and maintaining the best possible visual QoE for the first aid responder wearing the AR-HMD, while exploring packet loss and bandwidth throughput. The results of these experiments, involving latency, bandwidth aggregation, packet loss and FPS, are depicted in Figure 3 – Figure 6.

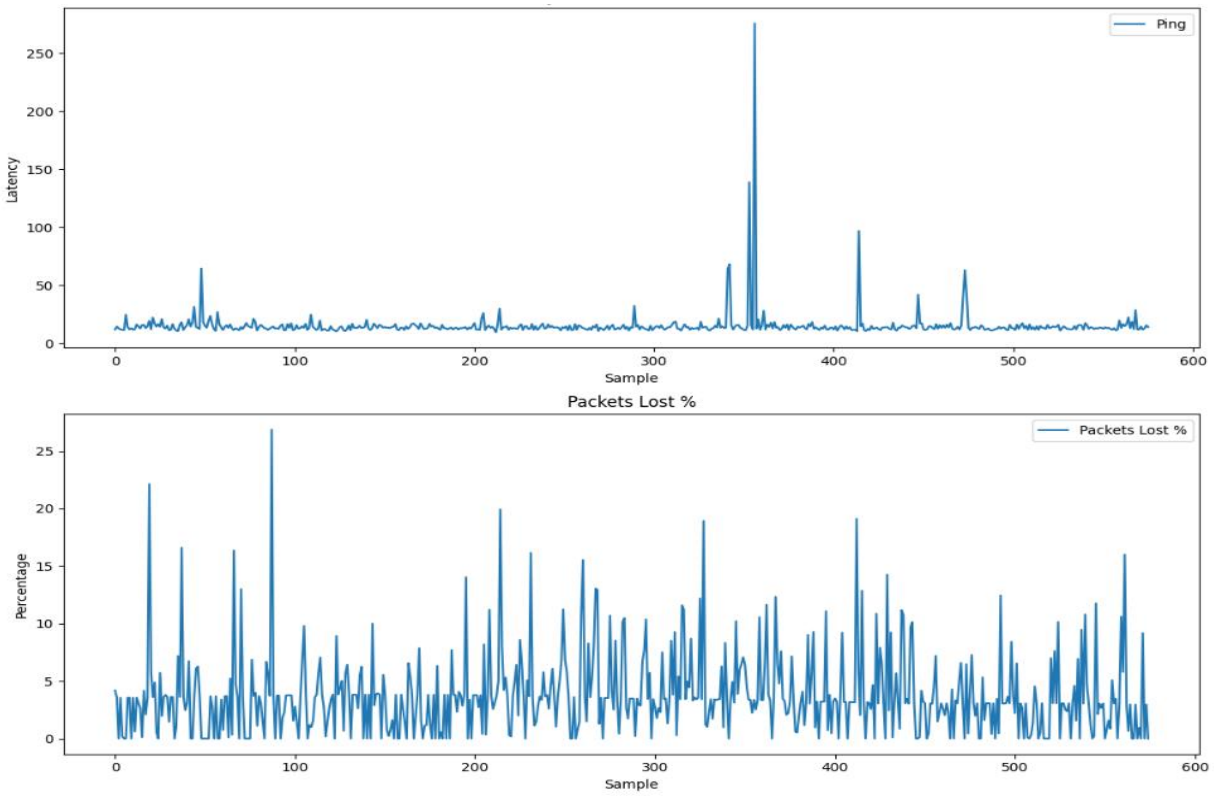

Figure 3: UC8 QoE-driven E1 experiment at CTTC (Latency & Packet Loss)

E1. QoE-driven Experiment (Lower Packet Loss): To achieve a pleasing, high-fidelity simulation for the AR user, characterized by a smooth video stream without image stuttering, or noticeable latency upon user actions, we employed a reduced bitrate of 50 Mbps (yielding an average of 50.37 Mbps) between the Application Server and the HMD. In this configuration, the video stream requires an average of 15.15 milliseconds to reach the HMD, taking advantage of the capabilities offered by 5G networks.

In this favourable encounter, participants experienced a heightened frame rate, approximately 60 FPS, devoid of discernible stuttering or lag in the video stream. The temporal interval between the experimenter's actions and the corresponding output in the rendered scene achieved minimal durations, thereby preserving user immersion. The evaluated packet loss was around 3.7 % on average.

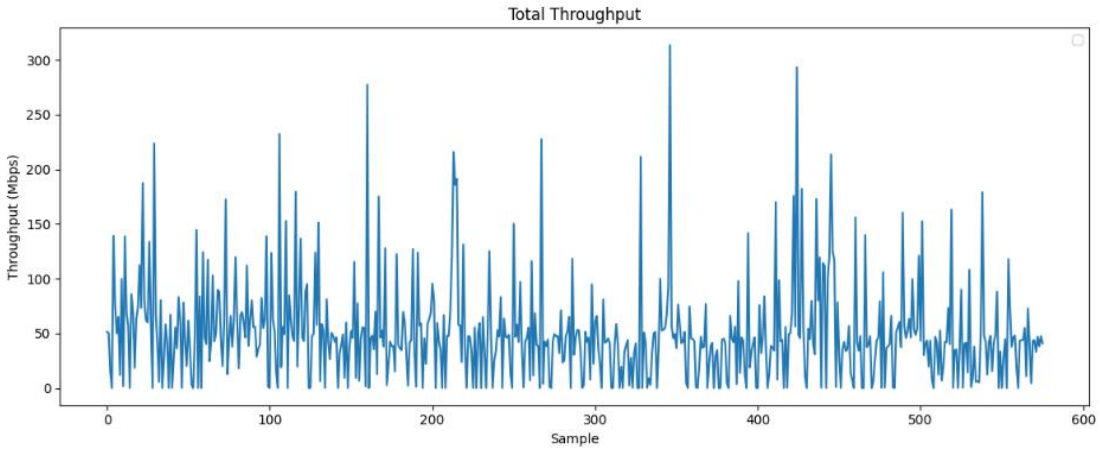

Figure 4: UC8 QoE-driven E1 experiment at CTTC (Throughput)

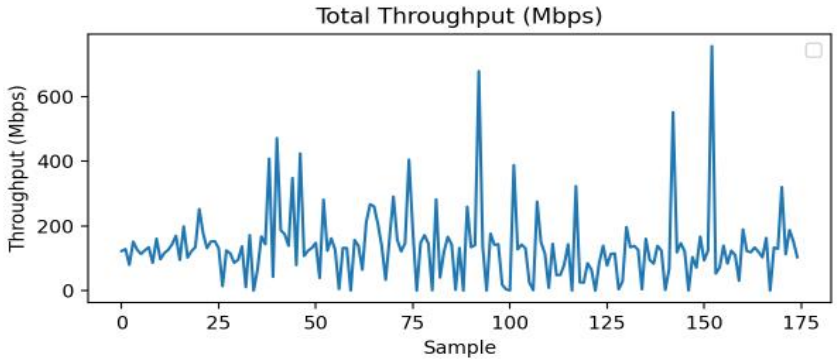

Figure 5: UC8 QoE-driven E2 experiment at CTTC (Throughput)

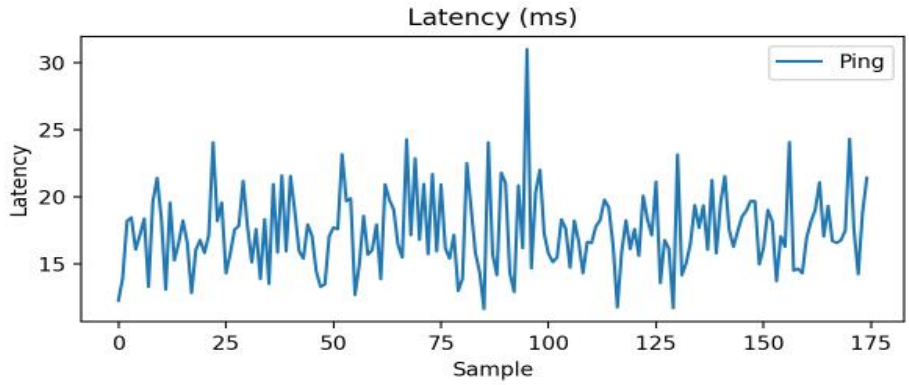

Figure 6: UC8 QoE-driven E2 experiment at CTTC (Latency)

E2. QoE-driven Experiment (Higher Throughput test): A second QoE-driven experiment was performed towards achieving a higher bandwidth throughput that would not compromise the user immersion and allow a smooth AR session. In this configuration, the video stream requires an average of 17.47 milliseconds to reach the HMD, taking advantage of the capabilities offered by 5G networks. An average throughput of 135 Mbps was sustainably achieved, with throughput exceeding 400 Mbps, sometimes spiking at more than 700 Mbps. A summary of the measured metrics is found in
Table 1.

Table 1: UC8 performance at CTTC

Experiment	Experiment Name	Target KPI	Performance
E1	Packet Loss Test	U ≥ 8% > A ≥ 4% > O	3.7%
E1	RTT Test	U ≥ 20ms > A ≥ 7ms > O	15.16 ms
E2	Throughput Test	U ≤ 0.4 Gbps < A ≤ 0.7 Gbps < O	0.135 Gbps

Lastly, we document the KPIs based on these experiments. Regarding battery consumption of the HMD, the original KPI of a 30% reduction is already documented, as the same principle, i.e., lower consumption due to remote rendering of the video, holds, independent of the choice of testbed. The actual vs. anticipated experimentation results for UC8 on the CTTC testbed are summarized in Table 2.

Table 2 : UC8 KPIs

KPIs	Results expected	Experimentation results
UC 8.1	E2E latency less than 7ms	E2E latency measured at 15.16ms (on average)
UC 8.2	Maximum aggregated total system bandwidth of at least 0.7Gb/s	Maximum combined overall system bandwidth of at 0.135 Gbps (maintained), sometimes exceeding 400 Mbps, and spiking at more 700 Mbps.
UC 8.3	Decrease device energy consumption by at least 30%.	Achieved 30% decreased device energy consumption.

Overall evaluation

The experimentation of UC8 at the CTTC testbed demonstrated that we can achieve a respectable QoE for the AR user, which is crucial for providing an immersive session. This result validates the ability of the 5G-EPICENTRE platform and the CTTC testbed to support immersive AR applications suitable for PPDR users. The packet loss of 3.7% on average was quite low, demonstrating the ability to serve applications that require such a low threshold to packet loss, without using the TCP protocol (which enforces 0% packet loss but induces latency and may create bottlenecks). The experiments at the CTTC testbed validated that AR applications requiring minimal packet loss and 5G-level round-trip times can also be supported. These results complement the experiments conducted at the UMA testbed that demonstrated that we can run AR applications with increased bandwidth aggregation.

CTTC platform conclusion

The CTTC 5G testbed is the experimental platform that in the framework of the 5G-EPICENTRE project is assigned to assess UC scenarios pivotal to understanding service instantiation and latency. In this regard, we employ an RTT tracker module to periodically monitor and publish the network RTT from the master node (of a Kubernetes cluster on a server) to the gNB and UE and published to RabbitMQ. RTT measurements were performed under normal

network conditions, where the gNB and the master node were set up in close proximity of each other, and the UE was positioned close to the gNB, approximately 2 meters away. The gNB and 5G Core were provided by Amarisoft ultimate, and the phone used was a SAMSUNG S22 with 5G SA support. The statistical values with a total of sixty samples for each RTT measurement route included a minimum of 0.418 ms to a maximum of 0.74 ms for the master node to gNB and 13.443 ms to 42.593 ms for the master node to UE, indicating that the low latency values obtained fulfil the requirements of the PPDR services of 5G-EPICENTRE, which are always in the optimal range except for the AR case, for which these values are in the acceptable range. The focus of the above configuration was on testing how far a regular/usual 5G configuration could go in fulfilling the PPDR requirements.

Further, to assess the service creation time, we developed a script designed to measure the time taken for service initiation. The script measures as starting time the instant in which the Helm deployment is started (through the corresponding command) within a specified Kubernetes namespace. Then, it monitors the status of the deployment and waits until all pods reach the "Running" state, signifying a successful deployment, at which point it calculates the total deployment time from initiation to completion. The results indicated that a sample complex PPDR service deployment of around 20 pods could achieve service creation times ranging from a minimum of 50 seconds to a maximum of 70 seconds, with a standard deviation of 9.24 seconds. This evaluation shows that the obtained service creation times are in general in the optimal range, and in any case, for the maximum measured values, they would still be in the acceptable range. Furthermore, the possibility of dynamically instantiating complex PPDR services next to the emergency spot in 10s of seconds is a feature that 5G networks bring to help PPDR offer a more efficient service in emergency scenarios. If we combine to this the capability to dynamically reallocate 5G network components in coordination with these services by adapting to network conditions, the advantages are further extended.

Conclusions

In conclusion, we have managed to reach each individual KPI under various settings: minimal end-to-end latency, high bandwidth aggregation, and a 30% decrease in energy consumption on the HMD. Additionally, we have validated that 5G-EPICENTRE can provide a platform where Unity-based applications, which are notoriously difficult to containerize efficiently, can be deployed and run. We have successfully deployed and run such an AR application using VMs, designed for the PPDR community, providing results that validate its efficacy. Our results further illustrate that, similar applications, requiring either minimal packet loss or high bandwidth aggregation, are supported by the 5G-EPICENTRE platform.

.

References

[1] C. A. Coventry, A. I. Vaska, A. J. Holland, D. J. Read, and R. Q. Ivers, "Surgical procedures performed by emergency medical teams in sudden- onset disasters: a systematic review," World journal of surgery, vol. 43, no. 5, pp. 1226–1231, 2019.

[2] T. Birrenbach, J. Zbinden, G. Papagiannakis, A. K. Exadaktylos, M. M ̈uller, W. E. Hautz, and T. C. Sauter, "Effectiveness and utility of virtual reality simulation as an educational tool for safe performance of covid-19 diagnostics: Prospective, randomized pilot trial," JMIR Serious Games, vol. 9, no. 4, p. e29586, Oct 2021. [Online]. Available: https://games.jmir.org/2021/4/e29586

[3] P. Zikas, S. Kateros, N. Lydatakis, M. Kentros, E. Geronikolakis, M. Kamarianakis, G. Evangelou, I. Kartsonaki, A. Apostolou, T. Birrenbach et al., "Virtual reality medical training for covid-19 swab testing and proper handling of personal protective equipment: Development and usability," Frontiers in Virtual Reality, p. 175, 2022.

[4] G. Papagiannakis, P. Zikas, N. Lydatakis, S. Kateros, M. Kentros, E. Geronikolakis, M. Kamarianakis, I. Kartsonaki, and G. Evangelou, "Mages 3.0: Tying the knot of medical VR," in ACM SIGGRAPH 2020 Immersive Pavilion, 2020, pp. 1–2.

[5] G. Papagiannakis, N. Lydatakis, S. Kateros, S. Georgiou, and P. Zikas, "Transforming medical education and training with VR using MAGES," in SIGGRAPH Asia 2018 Posters, 2018, pp. 1–2.