

Leveraging 5G Physical Layer Monitoring for Adaptive Remote Rendering in XR Applications

Inhar Yeregui*, Daniel Mejías*, Mikel Zorrilla, Roberto Viola

Fundación Vicomtech
Basque Research and Technology Alliance
San Sebastián, Spain
iyeregui@vicomtech.org

*PhD Candidate at UPV/EHU

Jasone Astorga, Eduardo Jacob

Department of Communications Engineering
University of the Basque Country
Bilbao, Spain
{jasone.astorga, eduardo.jacob}@ehu.eus

Abstract—As immersive eXtended Reality (XR) applications demand substantial network resources, understanding their interaction with 5G networks becomes crucial to improve them. This paper investigates the role of 5G physical-layer monitoring to manage and enhance the remote rendering of XR content dynamically. By observing network metrics directly from the physical layer, we propose a system to adapt streaming parameters such as bitrate, framerate, and resolution in real time based on available network capacity. Using theoretical formulas to estimate maximum data rate, our approach evaluates network resource availability, enabling the renderer to self-adjust media content representation. This is critical for providing consistent and smooth XR experiences to users, especially as network conditions fluctuate. Our findings suggest that physical-layer monitoring offers valuable insights to increase the Quality of Service (QoS) and has the potential to elevate user experience in remote-rendered XR applications.

Index Terms—Edge-based multimedia rendering, Physical layer monitoring, Extended Reality, QoS adaptation

I. INTRODUCTION

Extended Reality (XR), encompassing Virtual Reality (VR) and Augmented Reality (AR), is revolutionizing how users interact with digital environments. These technologies enable immersive experiences that blend the physical and digital worlds, offering applications in gaming, education, healthcare, and beyond. However, the resource-intensive nature of XR poses significant challenges, particularly when rendering complex content in real time. High-fidelity XR applications require substantial computational power to generate realistic 3D environments and process interactions at low latency. While some XR devices, such as high-end PCs or gaming consoles, can handle local rendering, many XR applications rely on remote rendering—offloading the computational workload to external servers or Multi-access Edge Computing (MEC) nodes. This approach enables lightweight XR devices, such as standalone headsets or mobile devices, to access high-quality experiences without the need for expensive onboard hardware. However, remote rendering introduces a strong dependency on network performance, as the rendered frames must be streamed in real time to the XR device with minimal latency.

Remotely rendered high-quality XR experiences depend on ultralow latency, high bandwidth, and consistent network

performance, making the interplay between XR applications and network infrastructure a critical research area [1]. The advent of 5G networks, with their reduced latency capabilities and enhanced bandwidth, has created new opportunities for streaming remote-rendered XR content efficiently [2]. Despite these advancements, network conditions in 5G environments can still fluctuate due to factors such as user mobility, network congestion, and environmental interferences. These factors can degrade both Quality of Experience (QoE) and the Quality of Service (QoS), manifesting as delays, lower resolution, or disrupted interactions.

Addressing this challenge requires a deeper understanding of the network behavior and adaptive mechanisms that can respond in real time to changes in network conditions [3]. This paper explores the potential of leveraging 5G physical-layer monitoring to tap into metrics such as signal strength, channel quality, and available bandwidth. By estimating network resource availability, XR streaming systems can dynamically adjust parameters such as bitrate, framerate, and resolution to enhance performance, ensuring a smooth and immersive experience.

The contributions of this paper are as follows:

- Utilizing physical layer monitoring to estimate available network resources: The paper demonstrates how monitoring specific metrics from the 5G network's physical layer, along with theoretical formulas to calculate data rate, can provide an accurate estimate of available network resources. This estimation allows the system to notice changes in connection quality and adjust to maintain a stable user experience in XR environments.
- Evaluation of the impact of adaptability on QoS: Through comparing the results to the ones obtained when no adaptability is implemented, this paper evaluates how auto-adaptation based on network resources can enhance the user experience by improving performance and reducing streaming issues in XR experiences. The research provides data on how controlling rendering quality parameters can significantly contribute to consistent QoS in variable connection scenarios.

The rest of the paper is structured as follows. Section II reviews related work on delivering immersive XR experiences over 5G solutions. Section III, focuses on the proposed physical layer-driven bandwidth estimation method, while its implementation in an auto-adaptive XR Remote Rendering system is described in section IV. Section V details the experimental assessment and finally, Section VI presents conclusions and outlines future work directions.

II. RELATED WORK

The convergence of XR technologies and 5G networks has become a critical research area in recent years. Several studies [4], [5], [6] have explored the unique challenges associated with delivering immersive XR experiences over 5G, focusing on network latency, bandwidth demands, and QoS. These works highlight the importance of developing adaptive mechanisms to address fluctuations in network conditions and ensure seamless user experiences.

Research on 5G networks has emphasized their potential to support low-latency, high-bandwidth applications. Authors in [7] have highlighted how 5G and 6G architectures can be leveraged to mitigate latency issues for XR, enabling interactive and immersive applications. However, this work also identifies challenges such as user mobility, which can disrupt the reliability of XR streaming.

The role of adaptive streaming systems in XR has also been the focus of several studies. Studies such as [8] and [9] investigated AI/ML-driven solutions for dynamic resource allocation in XR applications, showcasing the benefits of predictive models for real-time adjustments in bitrate and resolution. Similarly, in [10] a loopback mechanism is proposed to monitor and optimize network performance, demonstrating its ability to enhance QoS under fluctuating conditions.

Other contributions, like [11] have explored how XR-aware next-generation networks, spanning from the physical layer to the applications, enable the optimization and resource allocation to meet the demanding QoS requirements of XR services.

Despite these advancements, most solutions rely on general optimization frameworks, leaving a gap in XR-specific strategies that integrate physical-layer insights into adaptive streaming.

While the current body of research addresses critical aspects of XR performance over 5G, significant challenges remain. Many existing solutions lack scalability for dense user environments or fail to account for the variability of 5G networks in mobile scenarios. Moreover, few studies have explicitly focused on leveraging physical-layer monitoring to predict resource availability in real time.

This paper builds upon these studies by proposing a novel approach that utilizes physical-layer metrics to enhance XR streaming systems. It focuses on calculating the theoretical maximum data rate of a given system based on 5G wireless networks. By leveraging this data rate estimation, the proposed approach enables XR systems to dynamically adapt streaming

parameters in response to real-time network conditions. By bridging the gap between network-level data and real-time adaptation, the proposed method offers a more robust solution for maintaining high QoE in dynamic 5G environments. This contribution addresses the limitations of prior work and paves the way for more reliable and immersive XR experiences.

III. PHYSICAL LAYER-DRIVEN BANDWIDTH ESTIMATION

In several 3GPP releases, different network components, e.g., Network Exposure Function (NEF) [12], are standardized to provide access to network information such as configurations and metrics. Accessing such information enables any application to estimate the current network capacity and improve its behavior. Moreover, [13] includes the specifications of the radio access capabilities required for 5G User Equipment (UE) to operate in 5G NR (New Radio) networks. Among others, it describes how to calculate the maximum theoretical data rates for both downlink (DL) and uplink (UL), considering modulation, coding schemes, physical resource blocks (PRBs), and network overhead. The document provides configurations for channels and carriers, including the number of PRBs available for different bandwidths and subcarrier spacing. It also defines requirements for modulation orders and coding efficiency. This specification plays a critical role in ensuring interoperability between devices and networks while guiding manufacturers, operators, and researchers in the development and deployment of 5G technologies.

According to the mentioned specification, the maximum data rate of a 5G network for a user can be calculated using the following equation:

$$\text{Data Rate (Mbps)} = 10^{-6} \cdot \sum_{j=1}^J \left(v_{\text{Layers}}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{\text{max}} \cdot \frac{N_{\text{PRB}}^{BW(j),\mu} \cdot 12}{T_s^\mu} \cdot (1 - OH^{(j)}) \right) \quad (1)$$

The parameters presented in the formula are:

- J : Number of aggregated component carriers in a band or band combination.
- $v_{\text{Layers}}^{(j)}$: Number of transmission layers for carrier j .
- $Q_m^{(j)}$: Modulation order for carrier j , i.e., bits per symbol. E.g., 2 for Quadrature Phase Shift Keying (QPSK), 4 for 16 Quadrature Amplitude Modulation (16-QAM), etc.
- $f^{(j)}$: The scaling factor. It can take the values 1, 0.8, 0.75 and 0.4.
- R_{max} : Maximum code rate (e.g., $R_{\text{max}} = \frac{948}{1024}$).
- $N_{\text{PRB}}^{BW(j),\mu}$: Number of PRBs available for carrier j in the given bandwidth BW and numerology μ .
- T_s^μ : Symbol duration in seconds, which depends on the subcarrier spacing (e.g., for $\Delta f = 15$ kHz, $T_s = 71.4 \mu\text{s}$).
- $OH^{(j)}$: Overhead for carrier j , representing the fraction of resources not used for data (e.g., $OH = 0.14$ for frequency range FR1 for DL).

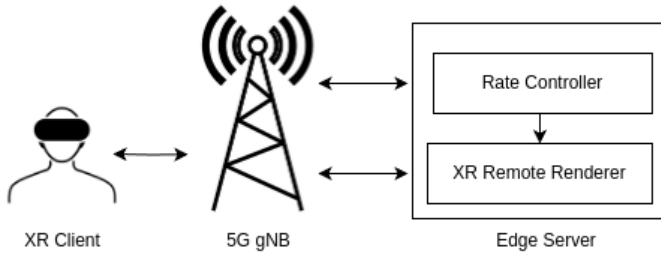


Fig. 1: Architecture for auto-adaptive XR Remote Rendering.

The information derived from the equation can be directly applied to a content streaming system with adaptive bitrate mechanisms. By periodically calculating the maximum theoretical data rate available on the 5G channel, the system can dynamically select the most suitable streaming profile based on current network conditions.

IV. ADAPTIVE REMOTE RENDERING FOR XR

Figure 1 shows the general architecture of an auto-adaptive XR Remote Rendering system. It consists of two main modules: the Rate Controller and the XR Remote Renderer. The Rate Controller is responsible for collecting network metrics and conducting bandwidth estimations that are going to be the input for the XR Remote Renderer to adapt the content streaming. Then, an XR Client receives the rendered media content through the 5G network.

The communication between the various components of the system is illustrated in Figure 2. The process of physic layer monitoring and the streaming quality profile updating is performed in a continuous loop. This cycle begins with the Rate Controller requesting the Modulation and Coding Scheme (MCS) index from the 5G gNB (1). The MCS index in a 5G network is a mechanism that determines how data is transmitted efficiently by combining modulation techniques and forward error correction coding rates. Modulation defines how digital bits are mapped to symbols, such as QPSK, 16-QAM, or 256-QAM, while the coding scheme adds redundancy to protect the data against errors. The MCS is dynamically adjusted based on the quality of the wireless channel; when conditions are favorable, higher-order modulation and lower redundancy allow for faster data rates, whereas in poor channel conditions, lower-order modulation with higher redundancy ensures more reliable communication, but reduce the achieved data rate. Upon receiving the MCS index, the Rate Controller estimates the wireless channel bandwidth using Equation 1 (2). If necessary, it communicates with the XR Remote Renderer to adjust the streaming quality profile (3). The XR Remote Renderer then updates the content streaming configuration accordingly (4). Concurrently, the XR Client receives the rendered video and audio streams and sends interaction data back to the XR Remote Renderer (5). Additionally, the XR Client exports performance-related metrics for further analysis (6). To prevent an excessive number of requests for quality changes to

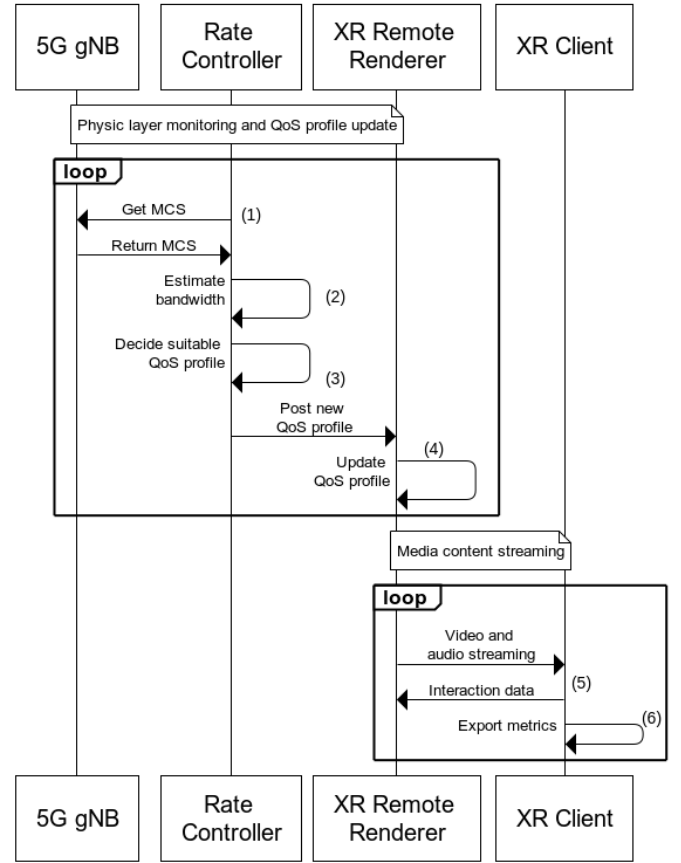


Fig. 2: Sequence diagram.

the XR Remote Renderer, a limit of one change every three seconds has been applied. This mechanism helps avoid unnecessary switching in scenarios where the channel quality fluctuates slightly above and below the predefined thresholds. By imposing this restriction, the system ensures stability and prevents rapid oscillations between streaming quality profiles due to minor, short-term variations in network conditions.

Three predefined streaming quality profiles are configured for the XR service, each defined by resolution, frame rate and bitrate [14] as described in Table I. The maximum data rate calculated every second determines the selected streaming quality profile, enabling the adaptation of the XR rendering quality to the available network capacity in real-time. The data rate ranges have been assigned to each quality profile taking into account the overhead that the RTP protocol adds to the data transmission [15].

TABLE I: Streaming Quality Profiles for Content Delivery

Profile	Resolution	Framerate (fps)	Bitrate (Mbps)	Data Rate Condition (Mbps)
Q1	720p	60	5	$data\ rate < 8$
Q2	1080p	60	8	$8 < data\ rate < 35$
Q3	4K	60	35	$35 < data\ rate$

V. EXPERIMENTAL ASSESSMENT

A. Testbed setup

To evaluate the theoretical data rate estimation and its impact on QoS in 5G networks, a testbed has been set up using the Amarisoft platform [16] to create a standalone 5G network. The Amarisoft software generates the 5G network, including both the core and radio access network (RAN) and provides detailed logs that include key network parameters. It has been configured with a specific network setup tailored to the requirements of the study. This configuration provides a controlled environment to evaluate the proposed methods effectively. The network parameters established in Amarisoft have been subsequently utilized to complete the bandwidth estimation equation. The key parameters of the testbed are listed in Table II.

TABLE II: Values used in the formula for Data Rate Calculation.

Number of Carriers (J)	1
Number of Layers (v_{Layers})	1
Bandwidth (BW)	40 MHz
Subcarrier Spacing (Δf)	30 kHz
Physical Resource Blocks (N_{PRB})	106 PRBs
Numerology (μ)	1
Scaling Factor (f)	1
Symbol Duration (T_s^μ)	$35.7143 \mu s$
Downlink Overhead (OH)	0.14
Sampling Rate (f_s)	1 second

The Amarisoft testbed allocates 106 PRBs, derived from the configured bandwidth and subcarrier spacing [17]. The Rate Controller processes the extracted MCS index, which determines the modulation order (Q_m), coding efficiency (f), and maximum code rate (R_{max}) as specified in [18]. In this case, the maximum data rate is calculated for downlink (DL), using equation 1.

The Rate Controller has been implemented using a Python script that communicates via WebSocket with the Amarisoft gNB to retrieve network quality metrics. After performing the estimation, it interacts with the XR Remote Renderer through an API, instructing it to adjust the streaming quality profile if necessary. The XR Remote Renderer has been implemented using the Unity framework [19] and is responsible for generating the 3D scene and rendering it for transmission to the XR Client using the WebRTC protocol. The XR Client is a web-based player implemented in JavaScript, capable of running on any device with a browser and WebXR support. Regarding deployment, both the Rate Controller and XR Remote Renderer are hosted on an Edge Server with a GPU, while the XR Client is deployed on a laptop equipped with a 5G modem. This testbed setup provides a controlled environment to evaluate the relationship between the calculated data rate, the network configuration, and the

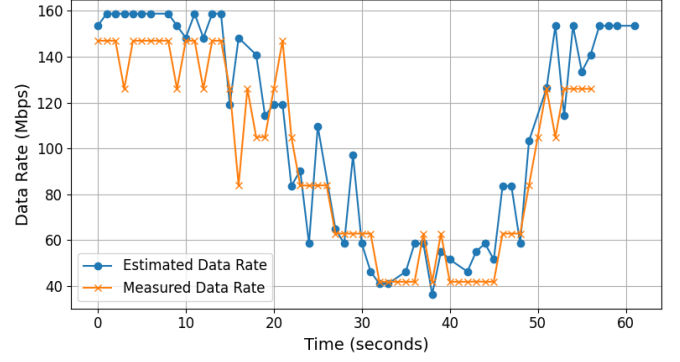


Fig. 3: Comparison between estimated data rate and measured throughput over time.

adaptability of streaming quality for a remote rendering XR service.

B. Results and discussion

To validate Equation 1, an experiment was conducted where the estimated data rate and actual throughput were measured simultaneously. The actual throughput was obtained using the *iperf3* tool [20], a widely used software for measuring network performance. The estimated data rate was calculated in real time using the parameters outlined in Section V-A, leveraging the data extracted from the Amarisoft logs.

Figure 3 shows the results of this validation experiment. The graph compares the estimated maximum data rate, as calculated using the theoretical equation, with the actual data rate measured via *iperf3* over a 60-second period. Both values were sampled at one-second intervals to ensure consistency in the comparison.

As shown in the figure, the estimated data rate closely follows the trend of the measured data rate, demonstrating the accuracy of the proposed equation under varying network conditions. While minor discrepancies can be observed, these are primarily attributed to dynamic network overheads, such as retransmissions and control signaling, which are not explicitly accounted for in the theoretical model.

This validation confirms that the proposed equation provides a reliable estimation of the available network capacity. Based on this, several experiments were conducted to observe the impact of the proposed auto-adaptive method on the XR service described earlier. To this end, QoS of a client consuming content generated by the XR remote renderer without any auto-adaptive quality method was compared to that obtained using the proposed auto-adaptive method.

To ensure a meaningful comparison of the impact of auto-adaptation, the evaluation was performed in a scenario where network capacity varied dynamically. Specifically, the test was designed so that the user equipment (UE), which runs the XR Client, moved along a predefined trajectory, starting next to the transmission antenna, where network conditions were optimal, and gradually moving away until reaching a maximum distance of 15 meters. The UE then returned

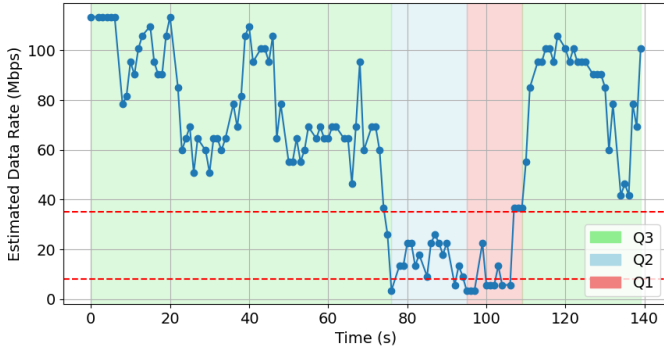


Fig. 4: Estimated data rate and current streaming profile.

TABLE III: Number of switches (S_{Nb}), number of freezes (F_{Nb}), total freeze duration ($F_{tot}(ms)$), average freeze duration ($F_{avg}(ms)$) and average framerate (fps_{avg}) for both No adaptation and Auto-adaptive Rate Control-enabled cases.

XR Remote Renderer	S_{Nb}	F_{Nb}	$F_{tot}(ms)$	$F_{avg}(ms)$	fps_{avg}
No adaptation	0	4	4760	1189	56.12
Auto-adaptive Rate Control	3	6	3930	786	56.51

to its original position, completing the trajectory in 140 seconds. This mobility pattern introduced controlled variations in network capacity, allowing the auto-adaptive method to be evaluated under fluctuating conditions.

Initially, the remote renderer was launched without any auto-adaptation mechanism, and various QoS-related metrics were collected. In this scenario, the transmitted multimedia content was configured to the Q3 quality profile, as described in Table I. The same test was then repeated with the auto-adaptive method enabled, ensuring statistical robustness through multiple iterations.

Figure 4 presents the estimated data rate during one of the tests conducted with auto-adaptation enabled. It also illustrates the streaming quality profile selected at each moment. In the graph, two horizontal lines can be seen, representing the data rate thresholds defined for selecting one streaming quality profile over another (8 Mbps and 36 Mbps). The changes in the streaming quality profile correspond to the variability of the estimated bitrate, which ranges from 158.796 Mbps (MCS = 27) to 5.025 Mbps (MCS = 0).

The observed metrics included the number of switches (S_{Nb}), the number of freezes (F_{Nb}), the total freeze duration ($F_{tot}(ms)$), the average freeze duration ($F_{avg}(ms)$) and the average framerate (fps_{avg}), and are presented in Table III.

In the case without adaptation, no changes in the streaming quality profile were observed, and four freezes occurred, resulting in a total freeze duration of 4760 ms. The average duration of each freeze was 1189 ms, while the overall average framerate throughout the test was 56.12 fps.

When applying the auto-adaptive method, three streaming quality profile switches were recorded during the test. In this case, six freezes were observed, with a total freeze duration

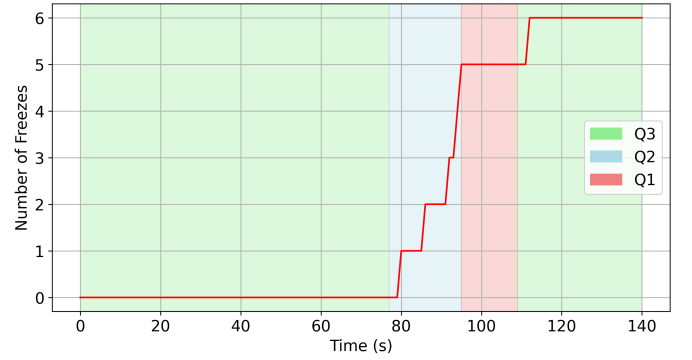


Fig. 5: Number of freezes and current streaming profile.

of 3930 ms and an average freeze duration of 786 ms. The average framerate remained relatively stable at 56.51 fps. The variation in framerate is due to the freezes, which cause the framerate to drop.

Comparing the results of both tests, we observe that the auto-adaptive method led to a higher number of freezes. However, the total freeze duration and the average freeze duration were lower than in the non-adaptive case. Regarding the framerate, no significant differences were observed. To gain a deeper understanding of the obtained results, additional tests were conducted using the auto-adaptive system. These tests revealed that every time the remote renderer performs a streaming quality profile switch, the WebRTC video streaming detects a freeze. The experiment was repeated ten times, confirming that each streaming quality profile switch consistently causes a freeze with an average duration of 277 ms. This behavior is attributed to a limitation in the remote renderer's implementation. Without this limitation, the number of freezes would be expected to be 3, and the total duration of freezes approximately 3099 ms.

Figure 5 shows the freezes and changes in the streaming quality profile over time. The graph illustrates that between seconds 70 and 80, QoS begins to degrade as freezes start to happen, until around second 100, which corresponds to the moment when the XR Client is farthest from the base station, causing the channel quality to worsen. During this period, the remote renderer adjusts the video streaming quality profile, lowering it from Q3 to Q2 and later to Q1. Subsequently, the remote renderer detects optimal network capacity again and returns to the Q3 quality profile, restoring streaming smoothness.

Taking into account the limitations of the XR service used in the experiments, it is observed that, during 76.43% of the time, the UE receives the content in the highest available quality. For the remaining time, the content is delivered at lower quality levels. This adaptation helps achieve a reduction of 17.4% in the total freeze time. Furthermore, if the aforementioned limitations, particularly those related to freezes caused by changes in the quality of the stream, were not present, the total freeze time could be reduced by as much as 34.9%. This demonstrates the potential for further optimization in the

system, should these constraints be alleviated.

VI. CONCLUSIONS AND FUTURE WORK

In this study, a theoretical model for estimating the maximum data rate of a 5G network has been utilized, and its potential application for enhancing the performance of a remote XR rendering service has been investigated. The findings demonstrate the positive impact of network bandwidth monitoring and estimation, combined with an auto-adaptive method to dynamically adjust the service's QoS according to the transmission channel capacity. The impact has been assessed by analyzing various metrics, including framerate, the number of freezes, and freeze duration. Furthermore, the results indicate that the observed performance is constrained by the implementation of the XR service employed in this study.

By bridging the gap between physical-layer network insights and XR application performance, this work lays the foundation for more resilient and adaptive XR systems. Future work will investigate how to leverage physical layer metrics to predict variations in network capacity, enabling auto-adaptation. This approach aims to further enhance system performance and quality of service.

ACKNOWLEDGMENT

This research was supported by the SNS-JU Horizon Europe Research and Innovation programme, under Grant Agreement 101096838 for 6G-XR project. It has also been partially funded by the Spanish National Plan for Scientific and Technical Research and Innovation, under award number CER-20231018, project 6GDiferente and by the national project Grant PID2022-137329OB-C44 funded by MICIU/AEI/ 10.13039/501100011033 titled "Enabling Native-AI Secure deterministic 6G networks for hyper-connected environments" (6G-INSPIRE).

REFERENCES

- [1] A. Clemm, M. T. Vega, H. K. Ravuri, T. Wauters, and F. De Turck, "Toward truly immersive holographic-type communication: Challenges and solutions," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 93–99, 2020.
- [2] W. Chen, X. Lin, J. Lee, A. Toskala, S. Sun, C. F. Chiasserini, and L. Liu, "5g-advanced toward 6g: Past, present, and future," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 6, pp. 1592–1619, 2023.
- [3] T. Taleb, A. Boudi, L. Rosa, L. Cordeiro, T. Theodoropoulos, K. Tserpes, P. Dazzi, A. I. Protopsaltis, and R. Li, "Toward supporting xr services: Architecture and enablers," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3567–3586, 2022.
- [4] M. Gapeyenko, V. Petrov, S. Paris, A. Marcano, and K. I. Pedersen, "Standardization of extended reality (xr) over 5g and 5g-advanced 3gpp new radio," *IEEE network*, vol. 37, no. 4, pp. 22–28, 2023.
- [5] H. Sampath, P. Tinnakornsriruphap, and P. Hande, "Enabling extended reality over 5g with distributed computing," *IEEE Communications Magazine*, vol. 62, no. 8, pp. 32–37, 2024.
- [6] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: Architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.
- [7] I. F. Akyildiz and H. Guo, "Wireless communication research challenges for extended reality (xr)," *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 1, pp. 1–15, 2022.
- [8] M. Laha, D. Roy, S. Dutta, and G. Das, "Ai-assisted improved service provisioning for low-latency xr over 5g nr," *IEEE Networking Letters*, 2023.
- [9] 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.
- [10] B. Bojović, S. Lagén, K. Koutlia, X. Zhang, P. Wang, and L. Yu, "Enhancing 5g qos management for xr traffic through xr loopback mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 6, pp. 1772–1786, 2023.
- [11] R.-A. Stoica, H. Bagheri, J. Löhr, and D. Karampatsis, "Xr-aware next generation networks," *IEEE Communications Magazine*, vol. 62, no. 8, pp. 62–68, 2024.
- [12] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "Mec in 5g networks," *ETSI white paper*, vol. 28, no. 2018, pp. 1–28, 2018.
- [13] "3GPP TS 38.306 V15.2.0 (2018-06): NR; User Equipment (UE) radio access capabilities," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 38.306, June 2018, release 15. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3289>
- [14] Google Support, "Recommended upload encoding settings," 2024, accessed: 2024-12-03. [Online]. Available: <https://support.google.com/youtube/answer/1722171?hl=en&zipy=%2Cvideo-codec-h%2Cbitrate>
- [15] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: A transport protocol for real-time applications," Internet Engineering Task Force (IETF), Tech. Rep. RFC 3550, 2003. [Online]. Available: <https://www.rfc-editor.org/info/rfc3550>
- [16] Amarisoft, "Amarisoft: 4G/5G Software Solutions," 2025, accessed: 2025-02-10. [Online]. Available: <https://www.amarisoft.com>
- [17] "5G NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone (TS 138 101-1 version 15.2.0 Release 15)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 138 101-1, June 2018, release 15. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3283>
- [18] "5G NR; Physical layer procedures for data (3GPP TS 38.214 version 15.2.0 Release 15)," 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 38.214, June 2018, release 15. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216>
- [19] Unity Technologies, *Unity Game Engine*, 2025, version 2025. Documentation available at <https://unity.com>.
- [20] ESnet (Energy Sciences Network), "iPerf3: A TCP, UDP, and SCTP Network Bandwidth Measurement Tool," 2025, accessed: 2025-02-10. [Online]. Available: <https://iperf.fr/>