Network Simulation for Investigating Latency Contributors in Collaborative AR Systems

BORA CIFTCI, University of Twente, The Netherlands

Focusing on collaborative augmented reality (AR) systems, where latency reduction is crucial for user experience and system performance, this study uniquely contributes by simulating real-world network traffic conditions using existing datasets. This research investigates the impact of network configuration parameters on end-to-end (E2E) latency in collaborative AR systems through detailed network simulation using the 5G RAN Emulator FikoRE. Our findings indicate that bandwidth, packet size, and the number of background users significantly affect E2E latency, with bandwidth emerging as the most critical factor. Future research directions include exploring the interplay of additional network variables and extending this work to more complex multi-user environments.

Additional Key Words and Phrases: Network Simulation, Collaborative AR, End-to-end Latency

1 INTRODUCTION

Augmented Reality (AR) technologies merge virtual elements with physical space, creating immersive experiences [1]. Collaborative AR systems, where multiple users interact within a shared AR environment, have the potential to transform many sectors by allowing for more natural and intuitive interactions among users. Collaborative AR systems are beneficial because they support teamwork, decision-making, and interactive learning [2]. By 2025, over 200 million people are expected to use extended reality services for immersive gaming and 95 million for live events [3].

Many companies have demonstrated significant interest in this technology by investing in AR. Microsoft, Meta, Google, and Nreal have significantly contributed to promoting AR glasses at the consumer level [2]. Vodafone has implemented an AR-based support system to enhance worker safety and operational efficiency by providing real-time data and guidance [4]. ETSI's ARCloud showcases the practical applications of collaborative AR in improving industrial processes by facilitating collaborative design reviews and factory inspections [5].

A significant challenge faced by these systems is end-to-end (E2E) latency, defined as the total time delay from the moment an action is initiated (e.g., a user moves their phone) to the perception of the corresponding response (e.g., the updated image) by the user. Latencies exceeding 20 milliseconds in collaborative AR systems can significantly increase the risk of cybersickness, characterized by symptoms such as nausea and dizziness [6]. Ensuring low latency is imperative for all immersive systems to provide a seamless and responsive user experience. Current AR systems aim for an end-to-end latency of approximately 10 to 15 milliseconds, depending on the required level of interactivity of the system [2]. The latency requirements become more stringent as AR systems become more dynamic and complex. For instance, running the YoloV4 Object Detection Model on a mobile device such as XiaoMi 9 requires a processing overhead of 286 milliseconds [7]. This leads to an E2E latency beyond the delay tolerable by the average user. Thus, minimizing latency is crucial to maintaining a seamless user experience [4].

Various studies have proposed practical solutions to mitigate latency in collaborative AR systems through offloading, utilizing mobile edge computing, and network slicing [4]. For example, the study in [3] provides an open framework for analyzing and modeling extended reality (XR) traffic by capturing packets transmitted by several virtual reality (VR) applications. Another significant contribution is the 5G RAN Emulator proposed in [8], designed to facilitate application-level testing for researchers. Additionally, [9] have provided a dataset derived from an offloading scenario for VR applications, which includes a model for generating synthetic data, and validated their results and tested several network parameters using the 5G RAN Emulator proposed in [8]. However, these frameworks have not been explicitly utilized to model the network traffic for collaborative AR or multi-user VR applications. The downside of not modeling the network traffic for these applications is that it can lead to unpredictable performance and scalability issues. The lack of modeling hinders the ability to optimize network resources effectively.

This research addresses a critical gap, given that collaborative AR systems involve unique interaction dynamics. Unlike traditional AR systems, where a single user communicates with a server, collaborative AR systems encompass multiple users. Making the network susceptible to issues such as interference. This study aims to fill the gap in the research by investigating the impact of network configuration parameters on the E2E latency of a collaborative AR application scenario. By utilizing the 5G RAN Emulator [8] with real traffic traces provided by [9], we can simulate the network traffic of a collaborative AR environment and assess the E2E latency under varying network configurations. This approach allows for the visualization of the impacts of the chosen network parameters on E2E latency and the identification of the parameter with the most substantial effect on latency. The parameter with the largest impact on latency is then investigated to determine the trade-offs regarding network quality, as the parameter is used to reduce the E2E latency. Therefore, we can frame the research around the following objectives:

- Understanding the root causes of E2E latency: Identifying the primary factors contributing to E2E latency in collaborative AR systems.
- Identify the largest contributor to latency: Determining which network configuration parameter has the most significant impact on latency.

TScIT 41, July 5, 2024, Enschede, The Netherlands

- Analyze the impacts of the largest contributor: Exploring how the most significant contributor to E2E latency affects overall system performance.

To facilitate achieving these objectives, we structure the research around the following research questions:

(1) *What are the primary network configuration parameters contributing to E2E latency in collaborative AR systems?*
(2) *Which network configuration parameter is the largest contributor to E2E latency in collaborative AR systems?*
(3) *How does the largest contributing network parameter to E2E latency affect network channel quality metrics?*

The subsequent sections of this report review existing literature and studies that address E2E latency and network traffic simulation frameworks for collaborative AR systems. Furthermore, the methodology and the experimental setup used to answer the research questions are explained, an analysis of the gathered results is provided, and the findings are interpreted in the context of existing literature and their implications for developing collaborative AR systems.

2 RELATED WORKS

2.1 End-to-end Latency

End-to-end (E2E) latency is defined as the total time delay from the initiation of an action to the perception of the response. The impacts of E2E latency on quality of experience (QoE) is analysed in [6], highlighting that latencies exceeding 20 milliseconds causes symptoms such as nausea and dizziness for the users. Stepwise latencies in interactive mobile video applications pipelines were dissected in [10], emphasizing the need to optimize network delays to achieve seamless user experiences.

2.2 Offloading

Offloading is a promising solution to reduce latency in collaborative AR systems as it decreases the hardware requirements of the equipment used through decreasing the computational load of the device [11]. Latency-aware computation offloading in 5G networks can significantly improve the performance of edge computing. The study in [12] outlines a framework for latency-aware offloading, aiming to minimize the end-to-end latency by optimizing the offloading process, considering factors such as network conditions and the computational demands of the tasks.

2.3 Network Simulation

The field of network latency in AR systems has seen significant advancements, with several studies providing frameworks and models crucial for understanding the latency dynamics of these networks. Multi-access edge computing (MEC) provides computing capability at the network's edge, which allows for offloading part of the computation from the utilized device to reduce network latency. An efficient and comprehensive deployment simulation framework is presented in [13] to test MEC deployments. However, the study focuses on vehicle-to-grid networks and does not address collaborative AR network requirements.

Realizing the lack of network simulation tools for XR traffic, [14] presents a novel traffic model for Virtual Reality (VR) applications based on real traffic traces from commercial VR streaming software. In [3], they extend the model to a framework for analyzing and modeling extended reality (XR) network traffic. They address the need to characterize and emulate all the information sub-flows to increase the fidelity of the traffic model.

To allow application-level experimentation and prototyping for 5G networks, [8] introduces FikoRE, a real-time Radio Access Network. Providing a tool to study the behavior of these networks under different network configurations to optimize the performance of specific applications. The real-time operation capabilities and the modular design make it an essential tool for testing and improving network configurations in real-world scenarios. Advanced XR applications rely on high-end hardware and next-generation wireless systems such as 5G and 6G. A detailed dataset and XR offloading IP traffic models have been provided in [9]. The dataset includes comprehensive traffic traces to enhance understanding of XR traffic dynamics. However, the study only considers one user generating the traffic acquired from real traffic traces and does not model a multi-user scenario required by collaborative AR systems. This dataset can provide realistic traffic patterns for simulation to model a collaborative AR system.

3 METHODOLOGY

This research utilizes the XR offloading dataset provided in [9] with the 5G RAN emulator FikoRE [8] set up as a simulator to use the real traffic traces captured from an offloading scenario. The simulation is configured to mimic the network traffic of a two-person collaborative AR system. We investigate the impacts of various network parameters on the E2E latency of a collaborative AR system using real network traces with a reliable 5G emulator. After calculating the E2E latency for all investigated parameter configurations, we determine the most significant contributor through linear regression analysis and analyze the associated network quality trade-offs.

3.1 Simulation Framework

The network simulation framework used in this study integrates FikoRE, a real-time 5G RAN emulator [8], with traffic models derived from the dataset provided in [9]. Utilizing real traffic traces allows for accurate modeling of network conditions and traffic patterns specific to collaborative AR systems. FikoRE was selected because its architecture aligns with the 3GPP specifications. Ensuring the framework accurately represents key procedures such as resource allocation and channel quality information metrics estimation. Adhering to 3GPP standards allows the simulation results to be comparable with real-world 5G deployments. Utilizing real traffic traces from an XR offloading scenario enables accurate modeling of traffic patterns specific to collaborative AR systems using an offloading solution.

FikoRE is a 5G RAN emulator that facilitates application-level experimentation and prototyping. The framework emphasizes usability and flexibility, allowing researchers to test their applications under various network configurations. Furthermore, the framework includes logging functionality to capture detailed information about

network performance. The logs include data on packet transmission, reception, resource allocation, and channel conditions.

The flow of data described in the FikoRE framework starts with simulated IP traffic being generated by the traffic generation module, generating virtual traffic based on user-specified parameters, such as target uplink (UL) and downlink (DL) throughputs and packet size. These packets are then forwarded to the PDC/RLC layer and queued in the IP buffer. These IP packets are then segmented into smaller blocks based on the Transport Block Size determined by the MAC layer. The MAC layer processes these segmented packets, performing resource allocation. Finally, the PHY layer receives the allocated resource blocks, estimates channel quality metrics such as SINR and RSRP using 3GPP 38.901 models and simulates transmission latency. An overview of the data flow is shown in Figure 1.

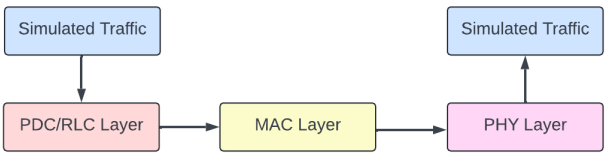

Fig. 1. **General Architecture of the FikoRE 5G RAN Emulator showcasing the data flow**: The flow starts with simulated IP traffic generated by the Traffic Generation module, processed through the PDC/RLC layer for queuing and segmentation, managed by the MAC layer for resource allocation, and evaluated by the PHY layer for channel quality and transmission parameters.

3.2 Data Collection and Analysis

3.2.1 Data Collection. After the simulation is run in any setting, FikoRE generates a log file for the user equipment (UE) module if the logging is enabled for the UE in the configuration file [8]. The generated log file is in text format. Some lines contain information about the data generated and received by the UE *(a)*. Other lines include details about channel quality metrics *(b)*. Each line includes the timestamp of when the log was taken, as shown in Figure 2.

The framework is designed to be compiled and run on Ubuntu, which is not the native OS of the host machine used for the experimentation. Therefore, a virtual machine running on Ubuntu runs the simulations and collects the log files. The network configuration parameters can be edited through the framework's configuration file. Collecting for each parameter setting involves editing the configuration file in the virtual machine and compiling and running the simulation. Then, the contents of the generated log files are copied and pasted into text files in the host machine. Finally, the E2E latency is analyzed, and the results are stored.

3.2.2 Data Analysis. To estimate the E2E latency, each instance of data generation by the user is timestamped, and the amounts of uplink and downlink data are recorded. The process starts by checking if there is any generated uplink (GUL) or downlink (GDL) data. If such data exists, a 'Generated Data' object is created and stored with its timestamp. When received uplink (RUL) or downlink (RDL) is detected, it is subtracted from the oldest 'Generated Data'

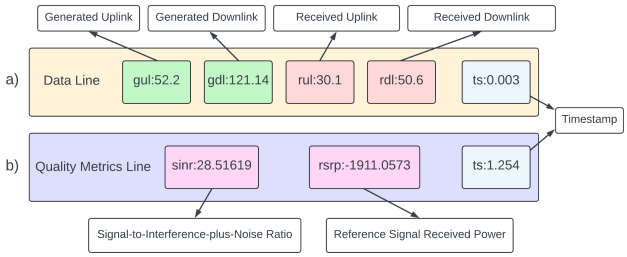

Fig. 2. **Contents of Log Files**: Section (a) features the information present in a data line (gul stands for Generated Uplink, gdl stands for Generated Downlink, rul stands for Received Uplink, rdl stands for Received Downlink, ts stands for Timestamp), and section (b) features information present in quality metrics line (SINR and RSRP values, ts stands for Timestamp) present in a log file collected from the simulations

object. Once the remaining data for an object is depleted, the latency is calculated as the time difference between data generation and reception, and the object is removed from the dictionary. The average latency is calculated from the list of latencies generated from the log file. If a line does not contain relevant data, it is skipped to ensure efficient processing. By timestamping each data generation event and tracking the uplink and downlink amounts, we can calculate the delay experienced during data transmission. The described approach accounts for processing delays and network latency and can be used to estimate the E2E latency.

Each run of the simulation results in slight deviations in the calculated latency. Therefore, the average of the latency measurements for ten runs is considered the average E2E latency for a specific parameter configuration. This approach ensures that outliers do not disproportionately affect the average E2E latency for any setting.

Experiments were conducted under five different settings for each parameter to measure their impact on average E2E latency. The relationship between the parameter and the average E2E latency is plotted to visualize the parameter's effect on latency. The parameter values are presented on the x-axis, while the average E2E latency is on the y-axis. Using line charts allows for the analysis of the consistency of the trends.

Linear regression analysis is used to identify the parameter that has the most significant impact on latency. This statistical method provides a list of coefficients that indicate the strength and the direction of the impact of each parameter on latency [15]. This method allows for a clear comparison of their relative importance. The linear regression equation used in this research is given in *Equation* 1. y is the dependent variable, representing the average E2E latency. x_1, x_2, x_3, x_4 are the independent variables representing the different network parameters (bandwidth, packet size, UE transmission power, and the number of background users). $\beta_1, \beta_2, \beta_3, \beta_4$ are the coefficients representing the impact of each network parameter on the average E2E latency. By performing linear regression analysis, the parameter with the highest absolute coefficient value (β) is identified as having the most significant impact on average E2E latency. Normalization is performed on the input parameters to

ensure values between 0 and 1. This allows for maintaining a consistent scale across different units and magnitudes of parameters. This approach is typically used in regression analyses to avoid bias toward larger-scale variables.

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \quad (1)$$

To analyze the trade-offs associated with utilizing the parameter with the most impact on average E2E latency, we also examine channel quality metrics SINR and RSRP. The SINR and RSRP values per configuration are determined by calculating the average of the values present in the log files. The data calculations are done for 10 runs per parameter configuration, and the average of the 10 runs is considered. This approach accurately measures the corresponding values for each configuration to visualize its impact on these metrics better.

4 EXPERIMENT SETUP

This section provides a detailed description of the experiment setup used to investigate the impact of various network configuration parameters on end-to-end (E2E) latency in collaborative augmented reality (AR) systems. The collaborative AR system we model in this research contains two users, which utilize the offloading XR traffic dataset to generate the network traffic. The scenario, depicted in Figure 3, mimics the network traffic of interacting within a collaborative AR application with another person using mobile phones.

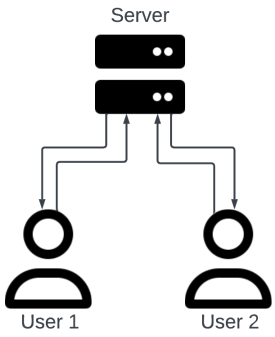

Fig. 3. **Modelled Scenario Connection Overview**: The figure illustrates the setup where two users interact in a collaborative AR application via mobile phones, with data offloaded to a central server for processing.

The FikoRE framework is utilized to model the specified scenario. The framework has been designed to be compiled and run on Ubuntu. Therefore, the simulations are run by a virtual machine (VM). The following section describes the specifications of the VM used to compile and run the simulations.

4.1 Simulation Environment

The network simulation was conducted using a virtual machine (VM), as the FikoRE simulator is designed to be compiled on Ubuntu. The use of a VM is necessary as the simulator is designed to be compiled on Ubuntu, which is not the native operating system of the host machine. The operating system deployed on the VM is Ubuntu version 24.04. The VM is allocated 8 GB of RAM, 6 processor cores, and 50 GB of storage.

The VM is hosted on a laptop running Windows 10. The host machine has 16 GB of RAM, 16 CPU cores, and 1 TB HDD. Half of the available RAM is given to the VM to ensure enough memory to run the simulations.

The following section provides a detailed description of the simulation parameters used to conduct the experiments.

4.2 Simulation Parameters

The simulation is executed with specific parameters kept constant, except for the parameter being tested, to ensure a controlled and systematic investigation. These values have been chosen to mimic the real-world conditions of a collaborative augmented reality (AR) system in which two users use their phones to interact within an AR application.

Collaborative AR systems require large data volumes and high data rates [2]. For example, interactions such as placing virtual objects or drawing virtual graffiti can result in high data transmission bursts, averaging around 2.5 MB per interaction. Continued user interactions can cause smaller data bursts, averaging 447 Bytes per interaction. These data patterns highlight the need for robust network infrastructure to efficiently handle large, unpredictable spikes in data transmission [16].

The bandwidth determines the data transmission capacity of the network. The chosen value for the bandwidth is 100 MHz, typical for high-performance 5G networks [5]. The packet size affects the transmission and processing time of data packets. Large packets can reduce the overhead but also increase transmission time due to the processing required to segment and reconstruct the packets [17]. A packet size of 12000 bytes is selected for the simulation to balance the trade-off between overhead and efficient data transmission.

The chosen modulation scheme is 256-QAM, which is appropriate for scenarios where high data rates are needed, such as in collaborative AR systems [9]. The frequency band is set to 26.5 GHz, typically used in 5G networks to provide ample bandwidth to support high data rates. The simulation duration affects the amount of data collected and the ability to observe the system's behavior. A duration of 30 seconds is sufficient to capture a representative sample of network performance metrics while ensuring that the simulation remains manageable and can be repeated multiple times for reliability. The transmission power available to the user equipment influences the signal strength and the coverage area. The selected transmission power of 23 dBm is typical for mobile devices [11].

Two UEs are simulated to reflect the typical use case for collaborative AR applications, where two users interact in real-time. The traffic file dictates the data patterns used in the simulation, which must reflect realistic usage scenarios. The selected traffic file for uplink and downlink transmissions for both UEs is the "960-offloading-real.txt" file provided by [9]. Although captured from an offloading scenario for VR applications, the traffic patterns are similar to the necessary transmissions for AR applications. Both cases require the video feed captured by the device to be sent to the

server, where computationally intensive algorithms such as localization and tracking can be performed, after which the results are transmitted back to the device.

4.3 E2E Latency Factors

The experiments are designed to analyze the impact of network parameters on end-to-end (E2E) latency in a collaborative AR scenario. The parameters were chosen after thoroughly reviewing the FikoRE framework's configuration options and carefully considering their impacts on E2E latency [4, 5, 6, 18, 17, 19], particularly in the context of collaborative AR systems [2, 4, 5]. The selected parameters for analysis are bandwidth, packet size, number of background users, and transmission power available to the user equipment.

In collaborative AR applications, high bandwidth is necessary to ensure a seamless and immersive experience [2, 4, 5, 6]. The bandwidth configurations (60, 80, 100, 120, 140 MHz) were selected to demonstrate the relationship between bandwidth and E2E latency. Varying these values significantly impacts latency and allows a more straightforward visualization of this relationship.

Data packets for collaborative AR systems carry complex and high-resolution information, which requires careful management to avoid latency spikes [18]. By examining various packet size configurations (8000, 10000, 12000, 14000, 16000 bytes), we can assess how packet size affects the E2E latency in a collaborative AR system.

The number of background users in a network can significantly impact latency due to increased competition for available bandwidth. Collaborative AR applications involve multiple users interacting simultaneously, leading to network congestion and potential delays [4]. The effect of the number of background users (1, 2, 3, 4, 5) present in the network is analyzed to understand their impact on E2E latency in collaborative AR systems.

Higher transmission powers can enhance the signal quality and reduce latency by improving the robustness of the data link. However, it also increases power consumption and potential interference with other devices [18]. By simulating various transmission power levels available to the user equipment (20, 30, 40, 50, 60 dBm), we can understand how it affects the E2E latency in a collaborative AR system. The chosen parameters and their respective configurations are presented in *Table* 1.

Table 1. The selected parameters (left column) and the corresponding configurations (right column)

Parameter	**Configurations**
Bandwidth	60, 80, 100, 120, 140 MHz
Packet Size	8000, 10000, 12000, 14000, 16000 Bytes
Amount of Background Users	1, 2, 3, 4, 5
Transmission Power of User Equipment	20, 30, 40, 50, 60 dBm

4.4 Software and Tools Overview

VirtualBox is used to host the virtual machine (VM), chosen for its ease of use and ability to configure the resources available to the VM. Visual Studio Code (VSCode) is used to connect to the VM through SSH to make changes in the configuration file. This approach also allows for copying and pasting the contents of the log files generated by the simulator onto the Python environment with relative ease. PyCharm, a popular IDE for Python, is used to analyze the contents of the log files. The contents of the log files can be copied onto text files directly from one IDE to another with the specified approach. Once all the necessary data is collected, linear regression analysis can be performed in the same directory used to analyze the log files. The Scikit-learn module for Python is used to perform the linear regression analysis. This module integrates machine-learning algorithms and linear regression analysis [20]. The Matplotlib module for Python is used to generate all the graphs presented in this research [21].

5 RESULTS

This section presents the data collected from the experiments through line charts and tables, aiming to visualize the impact of the parameters on average E2E latency. A table containing the coefficient values resulting from linear regression analysis for each parameter per user is given. The channel quality metrics are plotted with line charts to visualize the effect of the highest impact variable on overall system performance.

5.1 Visualization of Parameter Impact on E2E Latency

5.1.1 ***Bandwidth.*** Higher bandwidths provide more data transmission capacity, which reduces the time required for data packets to travel from the source to the destination [18, 17, 19]. The graph in Figure 4 visualizes the relationship between the available bandwidth and the average E2E latency in a collaborative AR system. The x-axis shows the bandwidth configurations and the y-axis shows the average E2E latency in seconds. The blue line corresponds to the first user's measurements, and the yellow line corresponds to the second user's measurements. As bandwidth increases from 60 to 140 MHz, the average E2E latency decreases.

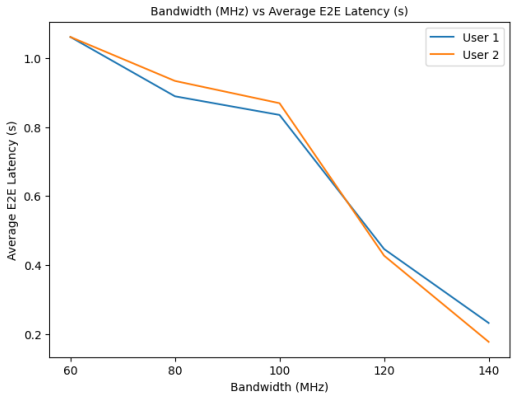

Fig. 4. **Bandwidth (MHz) vs. Average E2E Latency (s)**: x-axis represents the bandwidth in MHz; y-axis represents the average end-to-end latency in seconds. The blue line shows the measurements for the first user, and the yellow line shows the measurements for the second user.

5.1.2 ***Packet Size.*** The impact of packet size on average E2E latency is visualized through a line chart graph in Figure 5. The x-axis represents the packet size in bytes, and the y-axis represents the average E2E latency in seconds. The blue line shows the measurements for the first user, and the yellow line shows the measurements for the second user. As packet sizes increase from 8000 bytes to 16000 bytes, there is a noticeable increase in average E2E latency.

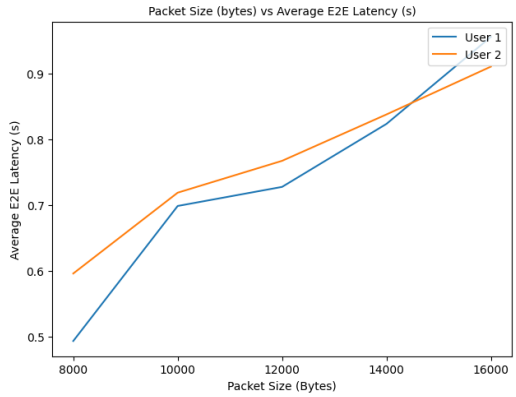

Fig. 5. **Packet Size (Bytes) vs. Average E2E Latency (s)**: x-axis represents the packet size in bytes; y-axis represents the average end-to-end latency in seconds. The blue line shows the measurements for the first user, and the yellow line shows the measurements for the second user

5.1.3 ***UE Transmission Power.*** The effect of the transmission power available to the user equipment on average E2E latency is shown in Figure 6. The x-axis represents the transmission power in dBm, and the y-axis represents the average E2E latency in seconds. The blue line shows the measurements for the first user, and the second line shows the measurements for the second user. As the transmission power increases from 20 to 60 dBm, the average E2E latency decreases.

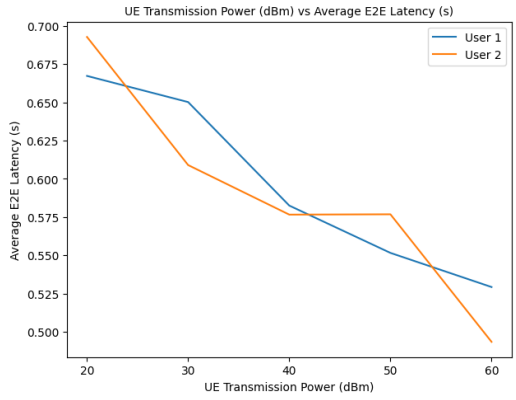

Fig. 6. **UE Transmission Power (dBm) vs. Average E2E Latency (s)**: The x-axis represents the transmission power available to the user equipment in dBm; the y-axis represents the average end-to-end latency in seconds. The blue line shows the measurements for the first user, and the yellow line shows the measurements for the second user.

5.1.4 ***Amount of Background Users.*** The effect of the number of background users on average E2E latency is visualized in Figure 7. The x-axis represents the number of background users, and the y-axis represents the average E2E latency in seconds. The blue line shows the measurements for the first user, and the yellow line shows the measurements for the second user. The average E2E latency increases as the number of background users increases.

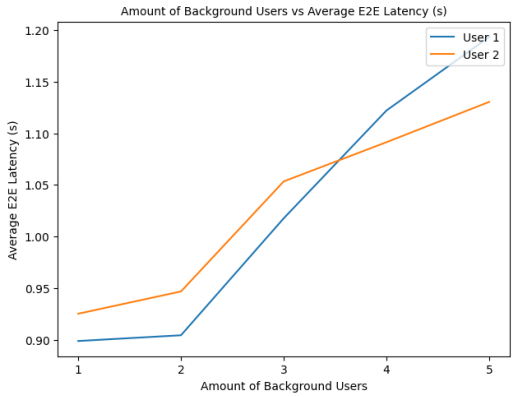

Fig. 7. **Amount of Background Users vs. Average E2E Latency (s)**: x-axis represents number of background users; y-axis represents the average end-to-end latency in seconds. The blue line shows the measurements for the first user, and the yellow line shows the measurements for the second user.

5.2 Linear Regression Analysis

The coefficients of the linear regression analysis quantify the sensitivity of E2E latency to changes in each network parameter for both users. Negative coefficient values indicate a negative relationship between the parameter and the average E2E latency. The absolute value of the coefficient is considered when deciding on the biggest contributor to ensure that the parameters with negative coefficients are not ignored. The coefficients resulting from linear regression analysis are presented in Table 2. The values are shown for both users per parameter. The results indicate the bandwidth has the highest absolute value for both users, making it the parameter with the most significant impact on E2E latency.

Table 2. **Coefficient values for both users per network parameter**: the left-most column presents the parameters, the middle column presents the calculated coefficient values for the first user, and the right-most column presents the calculated coefficient values for the second user.

Parameter	User 1	User 2
Bandwidth	-0.84	-0.91
Packet Size	0.42	0.30
UE Transmission Power	-0.15	-0.17
Amount of Background Users	0.32	0.22

5.3 Analysis of Network Quality Trade-Offs

In this section, we explore the observed relationship between the bandwidth and the network quality metrics present in the log files,

specifically, the Signal to Interference plus Noise Ratio (SINR) and Reference Signal Received Power (RSRP).

According to theoretical principles, increasing the bandwidth should decrease SINR. Higher bandwidth values introduce more noise power into the system. In contrast, RSRP, which indicates the power level of the reference signals received by the user equipment, should improve or remain stable with increased bandwidth, as it benefits from the broader transmission power distribution over a wider spectral range [22].

The RSRP values positively correlate with increasing bandwidth, indicating better signal strength and coverage. This aligns with our expectations. The relationship between bandwidth and RSRP is visualized in Fig. 8. Only one line is shown, as both users' measured RSRP values are the same. The x-axis represents the bandwidth in MHz, and the y-axis represents the RSRP values in dBm.

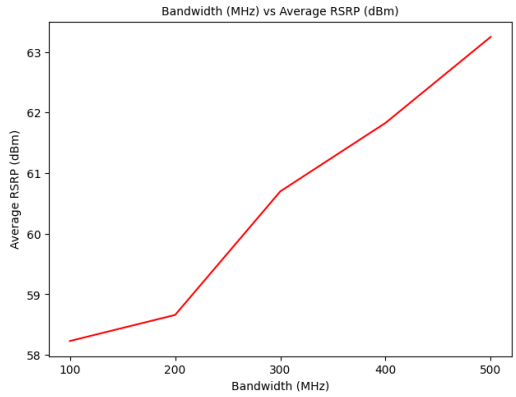

Fig. 8. **Average RSRP (dBm) vs. Bandwidth (MHz)**: The x-axis represents the bandwidth in MHz, and the y-axis represents the RSRP values in dBm.

However, the SINR values for both users remain at approximately 28.525 dB, disregarding bandwidth variations. This anomaly suggests that the FikoRE framework's model for SINR calculation may prioritize frequency band characteristics over bandwidth. Typically, SINR should degrade with increased bandwidth values due to the increased noise floor. However, the constant SINR implies a possible model limitation or an assumption within the framework that isolates SINR from bandwidth-induced noise effects. The relationship between the bandwidth and SINR is visualized in Fig. 9. The x-axis represents the bandwidth in MHz, and the y-axis represents the average SINR values in dB.

The implication of such behavior is critical for network planning and performance optimization. If SINR does not degrade with increased bandwidth, it may suggest that systems could leverage higher bandwidths without the expected compromise on signal quality, contrary to standard network theory. This could be beneficial in high-demand scenarios where high data rates (requiring large bandwidths) and robust signal quality (high SINR) are essential, such as collaborative AR systems.

Nevertheless, it is crucial to consider that this observation might be specific to the simulation environment and the assumptions embedded in the utilized framework. Real-world applications might still exhibit traditional behavior where SINR decreases as bandwidth increases.

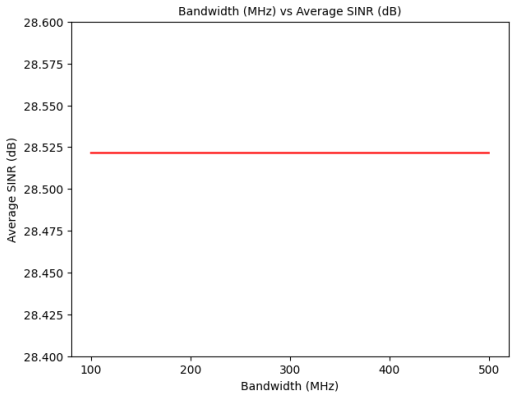

Fig. 9. **Average SINR (dB) vs. Bandwidth (MHz)**: The x-axis represents the bandwidth in MHz, and the y-axis represents the SINR values in dB.

6 DISCUSSION

This research provides concrete evidence of the varying impacts of network parameters on the average E2E latency for collaborative AR systems by leveraging a detailed network simulation environment.

Bandwidth was identified as the most significant contributor to average E2E latency. Higher bandwidths reduce latency by providing more data transmission capacity, which in turn facilitates faster data transfer between the user equipment and the server. This finding aligns with previous research indicating that sufficient bandwidth is critical for maintaining a seamless and immersive collaborative AR experience [2, 4, 5, 6].

Packet size also significantly impacted E2E latency, with larger packets increasing latency. This relationship can be explained by the longer processing and transmission times required for larger packets. While larger packets can reduce overhead by decreasing the number of packets that need to be processed, they also increase the chance of delays. Optimizing packet size is crucial to balance overhead and transmission efficiency.

The transmission power available to the user equipment influenced the E2E latency, with higher transmission power reducing the latency. Increased transmission power improves signal strength and transmission speed, reducing latency. However, this comes at the cost of higher power consumption and potential interference with other devices [23]. Therefore, while increasing transmission power can be a viable strategy for reducing latency, it must be balanced against the trade-offs to ensure efficient power usage and minimal interference.

The number of background users significantly impacted latency. This can be explained by the increased competition for available bandwidth. As the number of background users increased, the E2E latency also increased. This highlights the challenge of network congestion in collaborative AR systems. Emphasizing the need for efficient network management strategies to mitigate the effects of

congestion and maintain low latency in multi-user environments [4].

Analyzing the trade-offs associated with bandwidth revealed that while increased bandwidth improved signal strength (as indicated by RSRP values), the SINR values remained stable, which is unexpected. Typically, SINR values would be expected to decrease with increasing bandwidth due to higher noise power [22]. However, the stable SINR values suggest that the simulation environment may isolate SINR from bandwidth-induced noise effects, or it could be a limitation within the FikoRE framework. This finding is critical for network planning. It may indicate that higher bandwidth can reduce latency, but it might not always compromise signal quality.

Understanding the primary contributors to E2E latency allows for more targeted interventions to reduce latency. For instance, prioritizing bandwidth allocation and optimizing packet size can enhance system performance. Additionally, managing transmission power and mitigating network congestion are crucial for maintaining low latency in multi-user AR systems.

7 CONCLUSION

This research has demonstrated the substantial impact of network configuration parameters on end-to-end latency in collaborative AR systems. Through rigorous simulations using a 5G RAN Emulator and real-world traffic datasets, the primary factors contributing to E2E latency in collaborative AR systems are identified as bandwidth, packet size, transmission power, and the number of background users. It was determined that bandwidth is the most influential factor in minimizing latency., Increasing bandwidth reduced E2E latency by enhancing data transmission capacity. However, an unexpected finding was that SINR values remained stable despite the increase in bandwidth. This indicates that under the simulated conditions, higher bandwidth did not degrade signal quality, contrary to traditional expectations where increased bandwidth typically introduces more noise.

Future research directions can include exploring the interplay of additional network variables such as network slicing, quality of service (QoS) parameters, and more complex multi-user environments. Another direction could explore applying machine learning techniques to predict and mitigate latency in real-time. Extending this research to include various AR application scenarios, such as industrial AR and remote assistance, would provide a more comprehensive understanding of latency dynamics in different contexts.

REFERENCES

[1] Shaveta Dargan, Shally Bansal, Munish Kumar, Ajay Mittal, and Krishan Kumar. 2023. Augmented reality: a comprehensive review. *Arch. Computat. Methods.*, 30, 2, (Mar. 2023), 1057–1080. https://doi.org/10.1007/s11831-022-09831-7.

[2] Shuo Feng, Weiping He, Xiaotian Zhang, Mark Billinghurst, and Shuxia Wang. 2023. A comprehensive survey on ar-enabled local collaboration. *Virtual Reality*, 27, 4, (Dec. 2023), 2941–2966. https://doi.org/10.1007/s10055-023-00848-2.

[3] Mattia Lecci, Andrea Zanella Matteo Drago, and Michele Zorzi. 2021. An open framework for analyzing and modeling xr network traffic. *IEEE Access*, 9, (Sept. 2021), 129782–129795. https://doi.org/10.1109/ACCESS.2021.3113162.

[4] E. Stafidas and F. Foukalas. 2024. A survey on enabling xr services in beyond 5g mobile networks. *IEEE Access*, 12, (Apr. 2024), 59170–59197. https://doi.org/10.1109/ACCESS.2024.3392509.

[5] Theodoros Theodoropoulos et al. 2022. Cloud-based xr services: a survey on relevant challenges and enabling technologies. *Journal of Networking and Network Applications*, 2, (Jan. 2022), 1–22. https://doi.org/10.33969/J-NaNA.2022.020101.

[6] Maria Torres Vega et al. 2020. Immersive interconnected virtual and augmented reality: a 5g and iot perspective. *J. Netw. Syst. Manage*, 28, 4, (Oct. 2020), 796–826. https://doi.org/10.1007/s10922-020-09545-w.

[7] Jie Ren, Ling Gao, Xiaoming Wang, Miao Ma, Guoyong Qiu, Hai Wang, Jie Zheng, and Zheng Wang. 2021. Adaptive computation offloading for mobile augmented reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5, 4, (Dec. 2021), 1–30. https://doi.org/10.1145/3494958.

[8] Diego Gonzalez Morin, Manuel J. Lopez-Morales, Pablo Perez, Ana Garcia Armada, and Alvaro Villegas. 2023. Fikore: 5g and beyond ran emulator for application level experimentation and prototyping. *IEEE Network*, 37, 4, (July 2023), 796–826. https://doi.org/10.1109/MNET.002.2200595.

[9] Diego González Morín, Daniele Medda, Athanasios Iossifides, Periklis Chatzimisios, Ana García Armada, Alvaro Villegas, and Pablo Peréz. 2024. An extended reality offloading ip traffic dataset and models. *IEEE Transactions on Mobile Computing*, 23, 6, (June 2024), 6820–6834. https://doi.org/10.1109/TMC.2023.3326893.

[10] Teemu Kämäräinen, Matti Siekkinen, Antti Ylä-Jääski, Wenxiao Zhang, and Pan Hui. 2017. Dissecting the end-to-end latency of interactive mobile video applications. *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, (Feb. 2017), 61–66. https://doi.org/10.1145/3032970.3032985.

[11] Jacky Cao, Kit-Yung Lam, Lik-Hang Lee, Xiaoli Liu, Pan Hui, and Xiang Su. 2023. Mobile augmented reality: user interfaces, frameworks, and intelligence. *ACM Comput. Surv.*, 55, 9, (Jan. 2023), 1–36. https://doi.org/10.1145/3557999.

[12] Mobasshir Mahbub and Bobby Barua. 2021. Energy and latency-aware computation offloading and resource allocation for a multi-access edge computing-enabled heterogeneous network. In *2021 International Conference on Science Contemporary Technologies (ICSCT)*. (Aug. 2021), 1–6. https://doi.org/10.1109/ICSCT53883.2021.9642693.

[13] Yuankun Shi, Ziyang Peng, and Zhaojuan Bian. 2021. End-to-end behavior simulation for multi-access edge computing. In *2021 IEEE Asia Conference on Information Engineering (ACIE)*. (Jan. 2021), 65–72. https://doi.org/10.1109/ACIE51979.2021.9381092.

[14] Mattia Lecci, Andrea Zanella, and Michele Zorzi. 2021. An ns-3 implementation of a bursty traffic framework for virtual reality sources. In *Proceedings of the 2021 Workshop on Ns-3*. Association for Computing Machinery, (June 2021), 73–80. https://doi.org/10.1145/3460797.3460807.

[15] Gülden Kaya Uyanık and Neşe Güler. 2013. A study on multiple linear regression analysis. *Procedia - Social and Behavioral Sciences*, 106, (Dec. 2013), 234–240. https://doi.org/10.1016/j.sbspro.2013.12.027.

[16] Kittipat Apicharttrisorn, Bharath Balasubramanian, Jiasi Chen, Rajarajan Sivaraj, Yi-Zhen Tsai, Rittwik Jana, Srikanth Krishnamurthy, Tuyen Tran, and Yu Zhou. 2020. Characterization of multi-user augmented reality over cellular networks. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. (June 2020), 1–9. https://doi.org/10.1109/SECON48991.2020.9158434.

[17] Nisha Panwar, Shantanu Sharma, and Awadhesh Kumar Singh. 2016. A survey on 5g: the next generation of mobile communication. *Physical Communication*, 18, (Mar. 2016), 64–84. https://doi.org/10.1016/j.phycom.2015.10.006.

[18] Hao Yu, Tarik Taleb Masoud Shokrnezhad, Richard Li, and JaeSeung Song. 2023. Toward 6g-based metaverse: supporting highly-dynamic deterministic multi-user extended reality services. *IEEE Network*, 37, 4, (July 2023), 30–38. https://doi.org/10.1109/MNET.004.2300101.

[19] Zheng Ma, Ming Xiao, Yue Xiao, Zhibo Pang, H. Vincent Poor, and Branka Vucetic. 2019. High-reliability and low-latency wireless communication for internet of things: challenges, fundamentals, and enabling technologies. *IEEE Internet of Things Journal*, 6, 5, (Oct. 2019), 7946–7970. https://doi.org/10.1109/JIOT.2019.2907245.

[20] Fabian Pedregosa et al. 2011. Scikit-learn: machine learning in python. *J. Mach. Learn. Res.*, 12, (Nov. 2011), 2825–2830.

[21] Abdul Hafeez and Ali Sial. 2021. Comparative analysis of data visualization libraries matplotlib and seaborn in python. *International Journal of Advanced Trends in Computer Science and Engineering*, 10, 1, (Feb. 2021), 2770–2810. https://doi.org/10.30534/ijatcse/2021/391012021.

[22] Farhana Afroz, Ramprasad Subramanian, Roshanak Heidary, Kumbesan Sandrasegaran, and Solaiman Ahmed. 2015. Sinr, rsrp, rssi and rsrq measurements in long term evolution networks. *International Journal of Wireless Mobile Networks*, 7, (Aug. 2015), 113–123. https://doi.org/10.5121/ijwmn.2015.7409.

[23] Imtiaz Parvez, Ali Rahmati, Ismail Guvenc, Arif I. Sarwat, and Huaiyu Dai. 2018. A survey on low latency towards 5g: ran, core network and caching solutions. *IEEE Communications Surveys Tutorials*, 20, 4, (May 2018), 3098–3130. https://doi.org/10.1109/COMST.2018.2841349.