Evaluating the Impact of Mobile Network Quality on Robot Teleoperation Using a Remote Experience Center

Adam Janes[‡]*, Dong Yang[†]*, Jan Danek[‡], Xiao Xu[†], Zdenek Becvar[‡], and Eckehard Steinbach[†]
[‡] Dept. of Telecommunication Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague
[†] Chair of Media Technology, School of Computation, Information, and Technology,
Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany

***Abstract*—The rapid development of mobile networks opens new possibilities for collaborative robotics, particularly teleoperation, which requires highly reliable communication with low-latency. Nevertheless, researchers often face challenges due to limited cross-disciplinary expertise and restricted access to experimental infrastructure, such as advanced robotic platforms and 5G/6G testbeds. To address the fragmentation in teleoperation research, we present a modular experimental framework with graphical user interface (GUI), labeled as the Remote Experience Center (REC), which enables real-time robotic control over software-defined mobile networks. The REC integrates a mobile network testbed deployed in Prague with a robotic platform located in Munich. We demonstrate capabilities of the REC through a *pick-and-place* task involving human control with visual and force feedback. To improve safety and accessibility, the REC includes not only real hardware, but also digital twin of both the robotic system and the mobile network. Through real-world experiments, we investigate the impact of key mobile network parameters and characteristics, such as modulation and coding scheme (MCS), communication delay, and jitter, on user experience and system responsiveness. The experimental results reveal that increasing communication delay and jitter, as well as decreasing MCS, negatively affect the conditions for performing teleoperation tasks over a mobile network.**

***Index Terms*—Remote Experience Center, Software-defined Mobile Network, Real-world Experiment, Mobile Network, Digital Twin, Teleoperation**

I. INTRODUCTION

Teleoperation plays a key role in remote manipulation tasks in domains such as automation [1], [2], [3] or medical assistance [4], [5], [6]. The teleoperation over mobile networks has been the subject of extensive research in recent years, particularly with the rise of 5G. In a typical bilateral teleoperation systems with haptic feedback, control commands from the human operator are transmitted to the remote robot, while visual and haptic feedback is sent back to the operator, forming a closed loop over a communication network [7]. The quality of teleoperation is highly dependent on the performance of the underlying communication network, including key parameters such as delay, jitter, bandwidth, data bitrate or packet loss. Despite advances in 5G and the development of 6G, real-world cross-boarder teleoperation still remains challenge [8].

The mobile network limitations in teleoperation are addressed from three perspectives: i) control-side strategies, ii) communication-side enhancements, and iii) integrated approaches combining control and communication. From perspective of control-side strategies, works are focused on maintaining stability and transparency of the control loop under degraded communication conditions [7].

A variety of control-level strategies have been proposed, including the time-domain passivity approach (TDPA) and its extensions [9], [10], [11], [12], as well as model-mediated teleoperation (MMT) framework [13], [14], [15]. TDPA and MMT techniques aim to maintain robust teleoperation performance by compensating delay, jitter, and packet loss, through appropriate controller design, without modifying the communication network.

Communication-side works often use simulated mobile networks to study network impacts [16], [17], [18], but real-world mobile network introduces uncontrollable factors like interference and varying bitrate that strongly affect reliability and delay. Only a limited number of works evaluate teleoperation over real-word mobile networks [4]. However, mentioned work doesn't focus on dynamic resource allocation such as Physical Resource Blocks (PRBs) or Modulation and Coding Scheme (MCS), which directly influences how reliably low-latency bidirectional data can be maintained.

Integrated approach to address the limitations of mobile networks in teleoperation tasks involves a tighter integration of the robotics and mobile communication domains. However, this integration presents non-trivial challenges: roboticists often lack access to configurable mobile networks, while mobile network researchers rarely test on real robots. Recent works, such as [19] and [4], focus on bridging robotics and mobile communication. Nevertheless, these works do not address communication resource allocation aspects. To address this, we build upon the Remote Experience Center (REC) introduced in [19] and extend it with support for configurable resource allocation in both real and simulated mobile networks to investigate how resource allocation influences objective performance and subjective user experience.

---

*Adam Janes and Dong Yang contributed equally to this work and are co-first authors.

This work has been supported by the joint Czech-Bavarian research projects funded partly by the Bayerisch-Tschechische Hochschulagentur (BTHA/BAYHOST), Germany, and by the Ministry of Education, Youth and Sports, Czech Republic, under the projects no. BTHA-JC-2024-31 and LUABA24067, respectively.

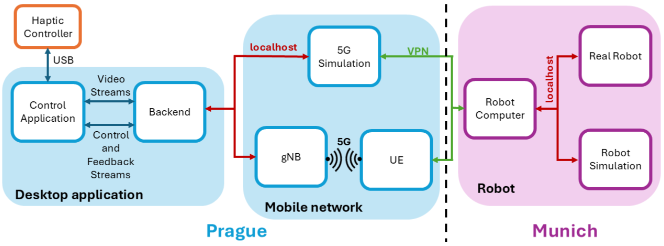

FIG. 1: Overall architecture of the REC framework divided into two sides (Prague and Munich) connected via VPN.

To address the gap in real-world evaluation and to support collaboration between researchers from both domains. The main contributions of this work are as follows:

1) We implement a modular and open-source framework that allows dynamic switching between real and simulated components on both the mobile network and robotic sides, enabling reproducible and flexible experimentation under various mobile network conditions.
2) We conduct a comprehensive evaluation of teleoperation performance over software-defined mobile network, combining objective task metrics and subjective user ratings to analyze how resource allocation and temporal network parameters affects user experience and task quality.
3) We make the REC framework enhanced with graphical user interface (GUI) and support for controlling mobile network parameters publicly available on GitLab[1].

In our implementation, a software-defined 5G mobile network and a haptic control interface are deployed in Prague, while the robot arm (telerobot) is located in Munich. The two sites are connected via a secure virtual private network (VPN) tunnel. The system supports both real and simulated versions of the robot and the 5G mobile network, enabling flexible evaluation of teleoperation performance under different mobile network conditions. Using the implemented setup, we analyze how wireless resource settings, such as MCS, delay, and jitter, influence objective and subjective metrics during a *pick-and-place* task.

The rest of this paper is organized as follows. Section II presents the system architecture detailing both the mobile network and the robotic system. Section III describes the experimental setup and outlines the robotic tasks used in the experiments. In Section IV, we present and discuss the results based on objective and subjective metrics. Section V concludes the paper and discusses future research directions.

II. SYSTEM ARCHITECTURE

This section first outlines the system overview and then provides detailed descriptions of the implementation of each component: i) the desktop application, ii) the mobile network with resource allocation capabilities, and iii) the robot simulation environment.

A. System Overview

The REC connects two research teams at geographically separated sites: one focusing on robotics and the other on telecommunications. The REC integrates a desktop application, a mobile network, and a physical robot or its Gazebo-based digital twin. In our setup, the control application, haptic controller, and mobile network is deployed in Prague, while the robot arm or its digital twin operates in Munich. Components in Prague and Munich communicate via a secure VPN, as illustrated in Fig. 1.

The desktop application provides GUI, see Fig. 2, for selecting the control device (mouse, keyboard, or haptic controller) and for adjusting key mobile network parameters in real-time, including maximum allowed MCS, maximum number of available PRBs, and additional delay or jitter. The selected settings parameters are sent via transmission control protocol (TCP) to the software-defined mobile network and applied accordingly. The user can also choose between the real software-defined mobile network with a Universal Software Radio Peripheral (USRP) or a Radio Frequency (RF) simulator[2]. In this work, the haptic controller is employed to ensure the most precise and intuitive control during the teleoperation tasks.

The mobile network is based on the open-source software-defined 5G mobile network platform OpenAirInterface (OAI) [20], with both real-world and simulated mobile network variants sharing the same containerized 5G Core (5GC). The gNB allows dynamic configuration of MCS and number of PRBs. In addition, we allow to add delay and jitter at the Internet Protocol (IP) layer using Linux kernel utility *tc qdisc*[3] and affect both uplink and downlink symmetrically. Decresing the MCS improves transmission robustness, but reduces mobile network bitrate. Adding delay and jitter simulate temporal instability of the mobile network degrading teleoperation controllability if excessive.

The uplink bitrate b_{ul} depends on modulation, coding rate, and number of allocated resource blocks [21], and is defined as:

$$b_{ul} = N_{ul}^{RB} \times N_{ul}^{sub} \times N_{ul}^{bits} \times N_{ul}^{sym} \times CR_{ul}, \qquad (1)$$

where N_{ul}^{RB} is the number of used resource blocks, N_{ul}^{sub} is the number of subcarriers per resource block, N_{ul}^{bits} is the number of bits per symbol, N_{ul}^{sym} is the number of symbols per subcarrier, and CR_{ul} is the code rate. The downlink bitrate b_{dl} is defined analogously with corresponding downlink parameters.

The final component of the REC setup is the robot or its digital twin, both located in Munich. The operator can control either the physical Franka Emika arm[4] or its digital twin within the Gazebo simulation environment[5]. Since this work primarily focuses on the networking aspects and some mobile network configurations could potentially result in unstable

---

[1] https://gitlab.fel.cvut.cz/mobile-and-wireless/remote-experience-center/resource-allocation

[2] https://gitlab.eurecom.fr/oai/openairinterface5g

[3] https://wiki.debian.org/TrafficControl

[4] https://robodk.com/robot/Franka/Emika-Panda

[5] https://classic.gazebosim.org/

system behavior, the simulation mode is selected for the experiments in order to avoid any risk of damage to the physical robot or its surroundings.

B. Desktop Application

The desktop application consists of a control application and a backend. The control application serves as the primary interface for the operator to configure key mobile network parameters (MCS, PRBs, delay, jitter) and to select between real or simulated mobile network as well as a real or simulated robot. The backend forwards data received from the control application and from the robot, see Fig. 1. The backend is connected to the control application via TCP and exchanges all control commands, video streams, and force feedback through a shared robot operating system (ROS) environment. The ROS environment is used to control either the Gazebo simulation or the physical robot.

The control application is implemented in HTML and JavaScript with Ultralight C++[6] as a lightweight rendering engine. Visual feedback from the robot or simulation is streamed directly to the control application. For precise teleoperation, the operator uses a Novint Falcon haptic controller [22], which provides continuous position/velocity input and receives force feedback commands from the desktop application to simulate haptic response.

C. Mobile Network with Resource Allocation Capabilities

The mobile network used in our experiments is based on the open-source software-defined 5G mobile network OAI [20]. We consider two deployment variants: i) the real software-defined network using a USRP B210 as the gNB and a Quectel RM520-GL 5G modem as the UE, and ii) the simulated network with emulated gNB and UE. Both variants share an identical protocol stack and mobile network parameters to ensure consistent and comparable experiments. The only difference lies in the transmission mode: while the simulation uses unacknowledged mode (UM), the real mobile network operates in acknowledged mode (AM). This difference arises from the fact that the hardware-based setup does not yet provide sufficient stability to enable reliable operation in UM for experimental purposes.

Both mobile network types run the USRP Hardware Driver (UHD) version 4.6 and the OAI gNB software built with the 2025.w11 tag. Version of the 5GC is 2.19 and 5GC is containerized using the Docker environment. The same 5GC instance serves both real and simulated configurations.

During experiments, the gNB and UE positions remain fixed. Communication uses 3.5 GHz band with 20 MHz bandwidth and 30 kHz subcarrier spacing. The PRBs are dynamically allocated according to traffic requirements and are hardware-limited to a maximum of 106 PRBs. The frame slot duration is set to 5 ms. The slot configuration within each frame consists of 6 downlink slots and 3 uplink slots.

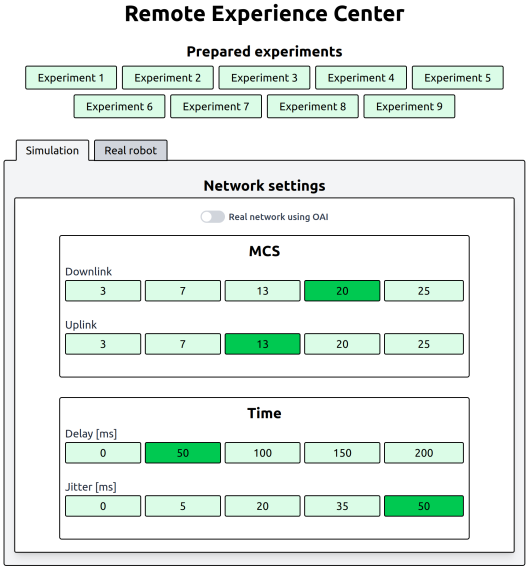

FIG. 2: Graphical User Interface of Desktop Application focusing on mobile network part.

Mobile network parameter adjustments (MCS, PRBs) are published via ROS, forwarded by a dedicated ROS node over TCP to a custom implemented OAI gNB module, which applies the settings. Delay and jitter modifications are sent via TCP to a Python script that configures the network interface using the *tc qdisc* tool.

To encapsulate the communication within the VPN and to enforce the use of the 5G mobile network, generic routing encapsulation (GRE) tunnels are employed. In our experiments, tunnels are established between Prague and the robot computer, between the robot computer and the 5GC, and between the robot computer and the UE.

D. Robot Digital Twin

In the REC setup, a Franka Emika robot arm with seven degree of freedom (DoF) serves as the follower, while the Novint Falcon haptic controller provides high-precise leader control with four DoFs (three translational axes and gripper manipulation). Both devices are chosen due to their accuracy and availability. Fig. 3 illustrates the communication between the desktop application and the Gazebo simulation.

The Gazebo environment hosts the robot's digital twin, which receives control commands from the desktop application. Inverse kinematics [23] compute the joint configurations, which are updated in the physics engine to calculate realistic force feedback based on inertia, friction, velocity, and environment interaction. This force feedback is sent back to the operator when haptic control is enabled.

Additionally, Gazebo streams two camera views (front and side) to the control application with a resolution of 1280 × 720 pixels at 30 Hz to enhance spatial awareness. Control commands and force feedback are exchanged at 1 kHz for smooth and responsive teleoperation experience.

III. EXPERIMENTS SCENARIO

We evaluate our framework using a classical teleoperation *pick-and-place* task executed under varying mobile network

---

[6] https://github.com/ultralight-ux/Ultralight

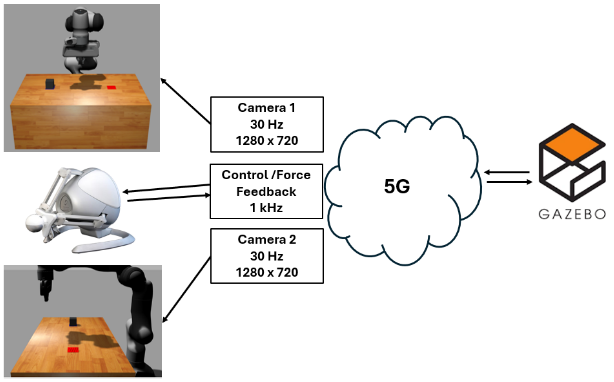

FIG. 3: Overview of the system illustrating data flow between desktop application and Gazebo simulation.

TABLE I: Experimental scenarios with maximum MCS and additional one way delay and additional one way jitter settings.

Experiment no.	1	2	3	4	5	6	7	8	9
Max MCS [-]	25	25	25	25	25	22	18	15	12
Add Delay [ms]	0	50	100	150	200	0	0	0	0
Add Jitter [ms]	0	5	20	35	50	0	0	0	0

TABLE II: Overview of MCS, number of bits per symbol, code rate, and theoretical downlink and uplink bitrates for a single resource block.

MCS	12	15	18	22	25
CR_{ul} and CR_{ul}	0.4238	0.6016	0.4551	0.6504	0.8027
N_{ul}^{bits} and N_{dl}^{bits}	4	4	6	6	6
b_{ul} / 1 RB [Mbps]	0.189	0.268	0.306	0.437	0.539
b_{dl} / 1 RB [Mbps]	0.378	0.537	0.611	0.874	1.078

conditions with both real-world and digital representations of the mobile network. The haptic feedback is computed using Gazebo physical engine, as detailed in Section II-D. To ensure consistency, the initial pose of the robotic arm and the position of the manipulated object are kept constant across all experiment runs.

As shown in Fig. 3, the *pick-and-place* task is performed in the Gazebo simulator using a digital twin of the Franka Emika robot arm. In both the real-world and simulated mobile network, the human operator is required to control the robot arm to pick up a designated object (i.e., the black cube, see Fig. 3) from a blue tray and place it onto a red tray. Visual feedback is provided by two fixed-position cameras. The first camera is positioned in front of the robot, offering a direct view of its movements. The second camera delivers a side view, enhancing the operator's perception of spatial depth, particularly along the x-axis of the robot's coordinate frame.

The experiment is designed to comprehensively evaluate the feasibility of teleoperation task using our REC framework under varying mobile network conditions and with both real-world and simulated mobile network. The primary objective is to assess the performance of cross-border teleoperation using the REC framework through both objective measurements and subjective user evaluations, thereby validating its robustness and usability in real-world scenarios.

A total of twenty participants take a part in the experiments. Ten participants perform the teleoperation task using a simulated mobile network, while the remaining ten use a real-world mobile network. Before the experiment, each participant is introduced to the objective and subjective metrics, the goal of the teleoperation task, and the operation of the platform, including how to control the robot's digital twin using the haptic controller. After this introduction, the participant is allowed several attempts to become familiar with the teleoperation task. Once the participant feels sufficiently prepared, he/she performs the teleoperation task nine times under different mobile network conditions, see Table I. After each run, objective performance metrics are recorded and the participant is asked to provide feedback on the subjective evaluation metrics. The order of the tasks is randomized and the participant sets the mobile network configuration using the GUI without knowing which specific network parameters are applied.

The evaluation metrics are divided into two categories: objective and subjective. The objective metrics include:

- *Time to Complete*: Measured from the task initiation to the completion (gripper release) of the *pick-and-place* task.
- *Precision*: Placement accuracy is quantified as the absolute 2D Euclidean distance between the center of the predefined target location and the actual center position of the black cube after completion of the *pick-and-place* task.

The subjective metrics include:

- *Force Feedback*: Perceived realism and responsiveness of haptic feedback.
- *Video Quality*: Perceived sharpness and fluid motion of the video stream.
- *Robot Control*: Ease of accurately controlling the robot.
- *Stability*: Consistency of mobile network connection without noticeable interruptions and drops in performance.
- *Usability*: Overall intuitiveness and ease of performing the *pick-and-place* task.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present and analyze the results obtained from the conducted experiments, focusing on both objective and subjective metrics to evaluate the impact of the mobile network parameters on teleoperation performance. Furthermore, this section highlights how the changes in the mobile network parameters affect the overall performance, user experience, and stability of the remote robotic control system.

Fig. 4 presents all subjective and objective metrics measured for the scenario with increased delay and jitter. Fig. 5 shows the same set of metrics for different maximum allowed MCS. The specific parameter values for both cases are listed in Table I and used code rates and number of bits per symbol as well as theoretical bitrate in uplink and downlink are detailed in Table II.

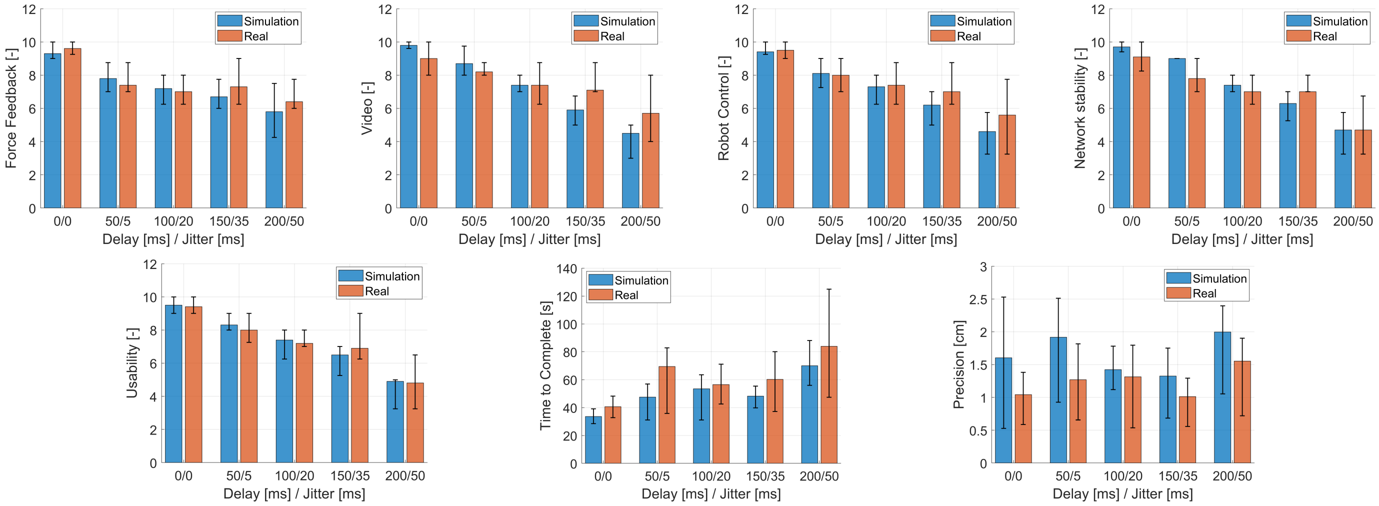

FIG. 4: Impact of additional delay and jitter on individual performance metrics. Displayed whiskers represent first and third quartile.

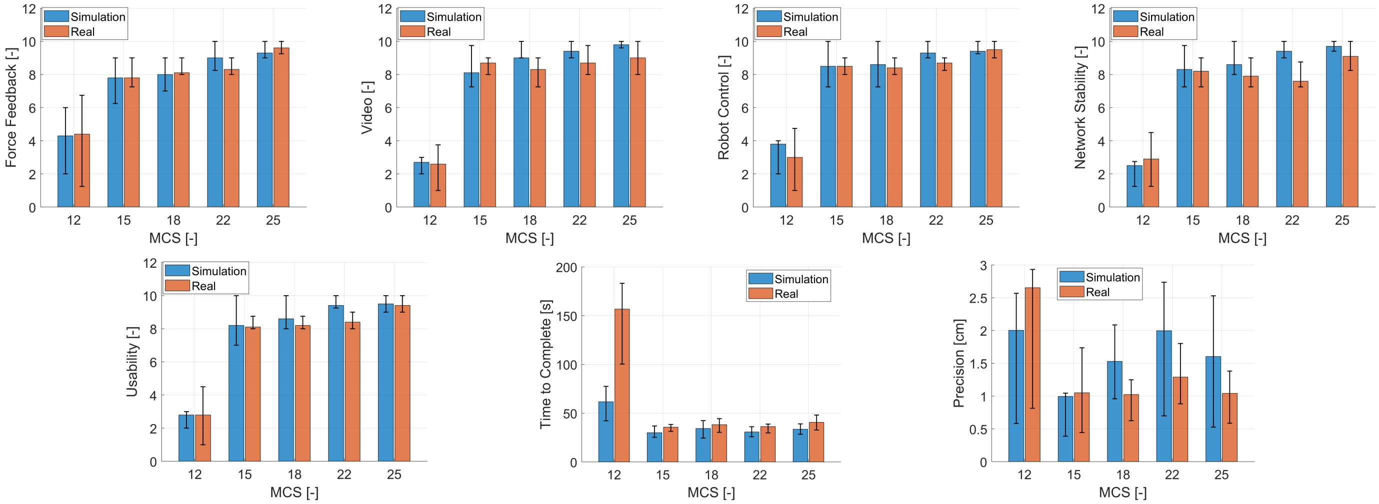

FIG. 5: Impact of MCS on individual performance metrics. Displayed whiskers represent first and third quartile.

The plots in Fig. 4 indicate that the scores of nearly all subjective metrics decrease as the mobile network delay and jitter increase. Conversely, the Time to Complete of the *pick-and-place* task rises with growing additional delay and jitter, as the increased delay complicates the execution of the teleoperation task. The plots showing the Precision of task completion as a function of additional delay do not exhibit any clear or consistent trend.

In Fig. 5, we observe that the scores of the subjective metrics show a very slight decreasing trend as the MCS decreases from 25 to 15. However, the lowest MCS value deviates from this trend and causes a notable drop in all subjective metric scores. Similarly, the Time to Complete of the *pick-and-place* task follows similar pattern as in subjective metrics, where a sharp increase in Time of Complete takes place for MCS 12. For the remaining MCS values, the Time of Complete remains almost constant. As with Precision depending on additional delay and jitter, the precision with respect to MCS does not exhibit any clear or consistent trend.

This unclear trend is most likely due to the nature of the task itself, where participants can anticipate the final placement accuracy just before releasing the robot's end-effector and deliberately adjust their actions to maximize precision. As a result, participants make conscious efforts to place the cube as accurately as possible on the tray, which is of comparable size to the cube. This focus on precision likely prevents any notable deterioration in the placement accuracy even under degraded mobile network conditions. However, this emphasis on precision contributes to the observed increase in the Time to Complete. If the participants are constrained by a fixed time limit (e.g., required to release the cube within one minute regardless of position), a decline in placement accuracy would likely be observed. We leave confirmation of this hypothesis for future research.

The obtained results are consistent with the initial expectations. As the additional delay and jitter increase, the teleoperation task becomes more challenging to perform, which is reflected by a longer Time of Complete and a rapid decline of all subjective metrics. The results focused on varying the MCS indicate a moderate reduction of the MCS value, which increases transmission robustness at the cost of reduced bitrate, has only a minor impact on the overall user-perceived quality as long as the network capacity remains sufficient to carry all necessary data streams. However, when the MCS is reduced to a level, where the bitrate becomes insufficient to transmit all required data (including control commands, force feedback, and two video streams) a significant deterioration in user-perceived quality occurs, accompanied by a considerable increase in Time to Complete of the *pick-and-place* task.

The difference in terms of gNB implementation between the simulated and real mobile networks is the mode of data transmission. In the simulated mobile network, it is possible to operate in the UM. In contrast, the real network requires the use of the AM due to limitations related to hardware handling. As a consequence of increased delay, jitter, and reduced MCS, transmission errors occurred, which manifest as out-of-order packet delivery. This effect is most apparent in the video streams, which are transmitted in UM across the mobile network and without packet acknowledgment, resulting in occasional frame skipping or sudden jumps between video frames. In contrast, the AM in the real mobile network ensures in-order packet delivery through the gNB, retransmitting any lost packets as necessary. This mechanism prevents frame skipping but instead results in occasional video stream freezing or stalling when packet retransmissions occurs. Despite these technical differences in packet handling, the results indicate that both modes have a nearly identical impact on the participants' perception of mobile network conditions during the experiments.

Apart from the difference in packet acknowledgment within the gNB, the underlying codebase used in both the real and simulated mobile networks is identical. The real mobile network operates with a USRP device, while the RF simulator emulated data transmission. From the results, it is evident that there is a strong correlation between the measurements obtained from the real mobile network and the simulated mobile network. This indicates that the RF simulator effectively replicates data transmission and accurately emulates real-world conditions.

V. CONCLUSION

In this paper, we have presented the REC framework extended with GUI and mobile network resource allocation for teleoperation over real or simulated mobile networks. The modular platform supports flexible switching between physical and digital twins of both the robot and the mobile network enabling reproducible testing under various mobile network configurations.

The results of experiments show that higher delay and jitter degrade user experience and increase Time of Complete. Reducing the MCS to increase transmission robustness initially causes only a slight decline in subjective quality. However, once the available bitrate becomes insufficient to handle control commands, force feedback, and dual video streams, the user's experience deteriorates abruptly and significantly. Additionally, the experiments confirm that the simulated mobile network closely replicates the behaviour of the real mobile network setup, demonstrating its usefulness for repeatable testing without requiring continuous hardware access.

In the future work, we plan to conduct a comparative analysis of PRBs and MCS configurations, including their combined effects on teleoperation performance under realistic error rates. We also aim to extend the platform with more diverse teleoperation tasks that demand higher precision and/or faster robot motion.

REFERENCES

[1] Z. Cao et al., "Adaptive Flow Scheduling for Teleoperation: A Communication and Control Co-Optimization Framework over Time-Sensitive Networks," IEEE Journal on Selected Areas in Communications, 2025.

[2] D. Yang et al., "ISSC: Interactive Semantic Shared Control for Haptic Teleoperation," 32nd IEEE RO-MAN, 2023.

[3] P. Meyer et al., "Web-Based Teleoperation System for Enhanced Precision and Efficiency in Food Industry Applications," 5th International Congress HORA, 2023.

[4] E. Babaians et al., "OCTOPUS: Optimized Cross-border TeleOperated Medicine Pouring Using NextGen Seamless Communication Networks," IEEE International Conference on Communications (ICC), 2024.

[5] P. M. Kebria et al., "HERCULES: Haptically-Enabled Remotely Controlled Ultrasound Examination System," IEEE Access, 2025.

[6] R. K. Megalingam et al., "Annapoorna–Food, Water and Medicine Delivery Teleoperated Robot with Telemedicine Facility," ICEMPS, 2024.

[7] M. I. Awan et al., "Model-Mediated Teleoperation for Remote Haptic Texture Sharing: Initial Study of Online Texture Modeling and Rendering," IEEE ICRA, 2023.

[8] K. Kalaiselvi et al., "Towards Seamless Connectivity: Implementing 6G Communication Technologies in Next-Generation Networks," 3rd ICONAT, 2024.

[9] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," IEEE Transactions on Robotics and Automation, 2002.

[10] J.-H. Ryu, "Bilateral control with time domain passivity approach under time-varying communication delay," RO-MAN 2007, IEEE, 2007.

[11] J.-H. Ryu, Q. Ha-Van and A. Jafari, "Multilateral Teleoperation Over Communication Time Delay Using the Time-Domain Passivity Approach," IEEE Transactions on Control Systems Technology, 2020.

[12] M. Panzirsch, et. al., "The 6-DOF implementation of the energy-reflection based time domain passivity approach," IEEE RA-L, 2020.

[13] P. Mitra and et. al., "Model-mediated telemanipulation," IJRR, 2008.

[14] Q. Deng, T. Faghanimakrani and A. H. Aghvami, "GBDT-Based Modules for Force Prediction in a Model-Mediated Teleoperation System," International Conference on Telecommunications (ICT), 2020.

[15] C. Kim and D. Y. Lee, "Adaptive Model-Mediated Teleoperation for Tasks Interacting With Uncertain Environment," IEEE Access, 2021.

[16] S. Liao and W.-k. Fung, "On network simulation for performance evaluation of real-time Internet-based teleoperation," IEEE International Conference on Robotics and Biomimetics, 2011.

[17] S. Acharya et al., "Network Emulation For Tele-driving Application Development," International Conference COMSNETS, 2021.

[18] C. Brueckner et al., "Simulation of Communication Network Latency Effects on Vehicle Teleoperation," IEEE ITSC, 2023.

[19] D. Yang, A. Janes et al., "Advancing Collaborative Research in Communication and Robotics: Insights from Experiments in a Remote Experience Center," IEEE ICC Workshops, 2025.

[20] F. Kaltenberger et al., "OpenAirInterface: Democratizing innovation in the 5G era," Computer Networks, 2020.

[21] 3GPP, "5G mobile system architecture, TS 23.501," 3GPP Technical Specification, 2022.

[22] S. Martin, and N. Hillier, "Characterisation of the Novint Falcon Haptic Device for Application in Virtual Environments," ACRA, 2009.

[23] Y. He and S. Liu, "Analytical Inverse Kinematics for Franka Emika Panda – a Geometrical Solver," ICCMA, 2021.