Introducing a 6G-Enabled Multi-Connectivity Robotic Teleoperation Platform

Eric Mittag*, Robin Klose§, Jan Herbst*, Matthias Rüb*, Jan Petershans*, Christian König*,
Matthias Hollick§, and Hans D. Schotten*‡
*German Research Center for Artificial Intelligence (DFKI), Germany, E-mail: {firstname.lastname}@dfki.de
§Secure Mobile Networking Lab, TU Darmstadt, Germany, E-mail: {firstname.lastname}@seemoo.tu-darmstadt.de
‡RPTU University of Kaiserslautern-Landau, Germany, E-mail: {lastname}@rptu.de

***Abstract*—As advancements in wireless technologies address the critical demands in medical robotics, industrial automation, hazardous material handling, and disaster response, it is crucial to test real implementations of upcoming wireless technologies. This work presents a teleoperation test environment that seamlessly integrates diverse wireless communication technologies, supporting multi-connectivity without requiring modifications to the core application code utilizing Robot Operating System 2 (ROS 2). This flexible testbed is used for the evaluation of network performance in terms of latency, packet delay variation (jitter) and reliability metrics. Additionally, the integration of a novel decentralized Ultra-Reliable Low-Latency Communication (dURLLC) system is demonstrated and evaluated, highlighting its potential as a reference setup for future communication technologies in 6G environments. Single-path and multi-path communication setups, including a dual 5G connection, are integrated into the testbed, compared, and evaluated regarding overall system performance. The results provide valuable insights into designing resilient and low-latency teleoperation systems, contributing to the advancement of 6G-enabled robotic applications.**

***Index Terms*—Teleoperation, Testbed, 5G, Beyond 5G, 6G, URLLC, ROS 2, Multi-Connectivity, Wireless Communication**

I. Introduction

With current work and research on beyond 5G or upcoming 6G wireless systems, various ambitious announcements are being made on its capabilities [1]. The proposed features include not only enhanced capacity and coverage, but also advanced capabilities such as integrated sensing and communication, AI-driven resource allocation, and secure by design architectures, all of which hold substantial promise for applications requiring stringent performance. Platforms for testing proposed communication features and technologies in realistic use cases are essential for the validation of theoretical models. In addition, they enable the evaluation of experimental setups under practical conditions and help mitigate uncertainties that may arise during the testing process. Robotic teleoperation has become a vital tool in mission-critical scenarios, including handling hazardous materials, disaster relief, and complex medical procedures [2], [3]. At the core of these applications is the need for precise control and real-time feedback, often enabled through haptic interfaces that convey force and tactile cues from remote robotic manipulators.

Using an advanced teleoperation system consisting of two Franka Research 3 collaborative robots (cobots) as executing systems, a customized testbed for potential 6G technologies is proposed. This setup can replicate real-world constraints while allowing researchers to systematically assess latency, reliability, and resilience. The bidirectional nature of the integrated haptic feedback adds complexity to wireless links. Beyond mere benchmarking, this platform also facilitates the investigation of new techniques for connectivity and data management in controlled settings. A key advantage of the teleoperation testbed is its ability to integrate and compare multiple wireless technologies.

Current frameworks frequently rely on single-path setups, which can be vulnerable to signal degradation, network congestion, and hardware malfunctions, or have only specialized use cases or measurement possibilities [4]. Multi-path communication distributes data streams across diverse network paths—such as Wi-Fi, 5G, and forthcoming beyond 5G links—reducing the likelihood of a single point of failure and improving overall robustness. When implemented at the application layer, as in this work using ROS 2 with its programmable nodes and underlying Data Distribution Service (DDS), multi-path approaches on link-level can be integrated with significantly less effort compared to solutions like multi-path TCP or Parallel Redundancy Protocol. This design strategy also simplifies adaptation to future wireless standards, enabling continual refinement of connectivity solutions without necessitating major changes to the underlying teleoperation platform.

The remainder of the work is structured as follows: Section II gives a short overview of related work in wireless testbeds, while Section III details the system setup and interfaces for wireless communication systems. In Section IV, the integration of a novel decentralized Ultra-Reliable Low-Latency Communication (dURLLC) system is presented, followed by experimental methodology in Section V. The results are discussed in Section VI, and Section VII summarizes the work and provides an outlook.

II. Related Work

In the development of upcoming wireless network technology, multiple testbeds have been presented in recent years. *Harjula et al.* present a smart manufacturing testbed featuring a 5G test network infrastructure including robotic applications [5]. The authors present their work on a hierarchical level without an example integration. In contrast, this work presents, in addition to communication standards such as 5G and IEEE 802.11, an integration of a custom dURLLC system.

Farnham et al. present a collaborative robotics testbed that, similar to this work, allows the integration of wireless communication technologies [6]. However, the focus is on swarm technologies, e.g. robot fleets in logistics with an orientation towards Internet of Things (IoT). Similarly, *Castillo-Sánchez et al.* present a testbed for wireless communication of robotic swarm systems [7]. Their approach is focused on low-end hardware on the robotic communication entities. Conversely, this testbed primarily connects two endpoints including human interaction and haptic feedback with hardware suited for more complex tasks.

Ayvaşık et al. present a haptic telemanipulator 5G testbed [8]. In contrast to this work, their setup consists of only a single robotic manipulator steered by a joystick implicating artificial scaling for virtually generated haptic feedback.

While different test environments as the ones addressed above target specific applications and communication protocols, there is a need for more flexible and realistic testbeds [4]. For example, an aspect not mentioned in the presented testbeds is the integration of multiple redundant wireless technologies between two endpoints and their impact on exemplary use cases. This shortcoming is directly addressed in this work presenting a setup built on top and enhancing a flexible cobotic teleoperation system formerly introduced in [9], with a focus on communication and protocol flexibility and real-world applications.

III. System Setup and Integration Interfaces

This section outlines the teleoperation system's hardware, software, communication technologies, and integration interfaces for seamless data exchange.

As illustrated in Figure 1, the teleoperation system consists of two Franka Research 3 cobots in a leader-follower configuration, along with supporting hardware for communication and control. The setup occupies an area of approximately three meters in length and width. On the left side of the setup, the novel dURLLC system is visible, demonstrating its integration into the teleoperation testbed.

Fig. 1: Physical Setup of the teleoperation system featuring two Franka Research 3 cobots in a leader-follower configuration and the dURLLC system

A. Teleoperation

The used teleoperation system builds on top of a ROS 2 enhanced version of [9], utilizing two Franka Research 3 cobots, each with seven axes and a control cycle frequency of 1 kHz, in a leader-follower configuration. This high frequency ensures precise acquisition of the cobot's state and supports direct torque control for real-time operation. Low-level communication with the cobots is managed by the libfranka library, providing access to measurement data and control commands from within the ROS 2 framework using the franka_ros2 package. The teleoperation code, originally based on Franka Robotics' implementation [10] running on one computer and ROS, has been divided into two controller instances for leader and follower, running also at 1 kHz, as described by *Petershans et al.* [9]. The setup used in the presented work is transitioned to ROS 2.

After alignment of the poses between the two cobots, the follower continuously mirrors the leader's joint states in real-time, while providing force feedback to the leader.

The system operates in two states: ALIGN and TRACK. In the ALIGN state, the follower robot moves all its joints to match the positions of the leader robot. In the TRACK state, the follower robot continuously mirrors the leader's joint positions, velocities, and efforts in real-time. From the leader to the follower, three topics are used: *joint_states* (position, velocity, and effort of each joint), *contact_scaling* (for force feedback computations), and the leader's initial joint states for the ALIGN phase. From the follower to the leader, three additional topics are employed: *aligned_flag* (indicating successful alignment), *contact_scaling* (also for force feedback computations), and *tau_ext_hat* (external forces applied to the follower).

During this teleoperation phase, data exchange between the leader and follower relies on three ROS 2 topics for each side. The data rates are measured here as 1.2 Mbit/s from the leader to follower and 0.7 Mbit/s from follower to leader at 1 kHz for the raw ROS 2 topics. For the DDS-based implementation, the rates increase to around 4 Mbit/s leader-to-follower and 3.5 Mbit/s follower-to-leader.

B. Software Stack

Based on the TCP/IP reference model, Figure 2 shows the structure of the system's software for data exchange. At the application layer, ROS 2 Humble is used for communication via Fast-DDS middleware, enabling real-time data transfer. Following, the standard Linux network stack is used for 5G and Ethernet via ModemManager and NetworkManager. The system runs on a preemptive real-time kernel (version 6.1.0-27-rt) to meet strict timing requirements. For portability, the entire setup is containerized using Docker. Each robot computer runs a Debian 12.8 host operating system with an Ubuntu 22.04-based container for ROS 2 processes. This allows modularization, easy deployment and testing of the teleoperation software across different hardware setups.

C. Communication Systems

The system supports multiple communication technologies to transmit data between the leader and the follower:

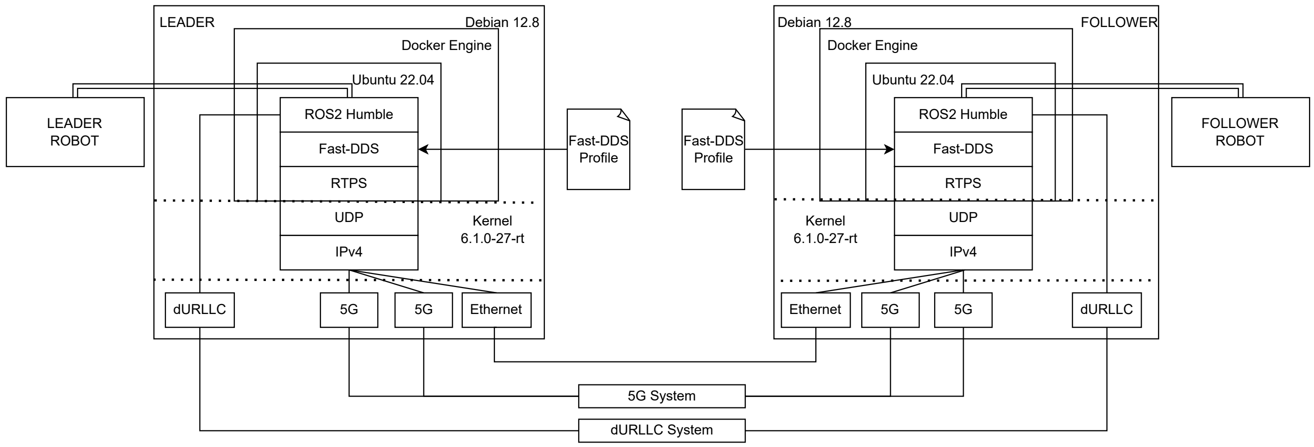


Fig. 2: Logical teleoperation system overview with integrated communication systems

i) Ethernet: A 10 Gbit dual-port Network Interface Controller (NIC), based on the Intel X540-T2 chipset, is used for baseline tests. It provides a high-reliability wired connection and synchronizes the whole system using the Precision Time Protocol (PTP).

ii) 5G System: A MECSware campus network operating on a 100 MHz bandwidth at 3.75 GHz with a Sercom SCE5164-B78 base station. As User Equipment two different Quectel modems (RM500Q and RM520N) are in use.

iii) dURLLC: This system is described in Section IV. Its integration showcases the capability of integrating and testing custom communication systems.

D. Communication System Integration Interfaces

The teleoperation platform supports two methods for integrating communication systems: default Linux networking and a custom ROS 2 node-based approach.

The Linux-based integration, used e.g. for Ethernet and 5G, employs the NetworkManager utility for managing connections through standard IP-based protocols. This straightforward method is ideal for technologies that natively support UDP communication. A Fast-DDS profile on the respective side adds these interfaces to the ROS 2 environment to ensure that only the desired interfaces are used.

The ROS 2 node-based integration provides greater flexibility for specialized systems. For example, the decentralized URLLC system uses a custom node to translate ROS 2 topics into a serialized UDP connection, consolidating all data into a single port in each direction. This simplifies the communication pipeline compared to the standard DDS setup to ensure compatibility with the external communication system. ROS 2 node-based systems can be controlled dynamically by starting or stopping specific nodes.

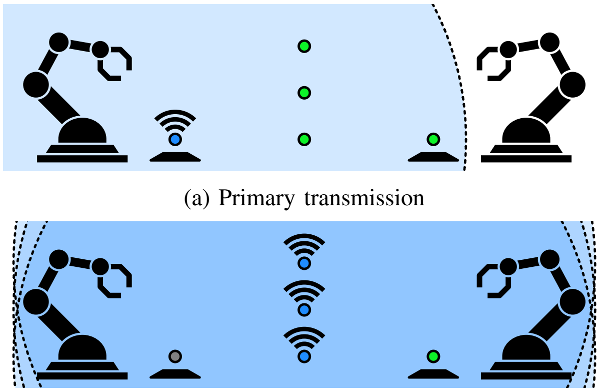
Duplicate messages, whether from redundant communication systems or other integrations, are handled at either the DDS or ROS 2 level. The system ensures that only the first-arriving message is used, maintaining reliability and low latency. These integration options enable seamless adaptation to both standardized and experimental communication technologies.

IV. Integration of Decentralized URLLC System

A novel communication system for decentralized URLLC [11] is integrated into the teleoperation testbed to showcase a custom technology integration and to assess the dURLLC system under functional interactions with the real-world robotic application. The dURLLC system essentially features fast and reliable data transfers over relay nodes through concurrent cooperative transmission (CCT).

In a CCT, multiple nodes transmit the same data at the same time under tight synchronization constraints such that their signals interfere without causing a collision, enabling a receiver to process the superimposed signal and to decode the data. The CCT communication scheme was popularized in the field of low-power wireless networks, where protocols for low-latency network flooding and cyber-physical systems were proposed [12]. However, these preceding solutions for low-power wireless networks are based on physical layers (PHYs) with rather low data rates, e.g., Glossy [13] uses IEEE 802.15.4 (250 kbit/s) and BlueFlood [14] uses Bluetooth (up to 2 Mbit/s). In contrast, the dURLLC system that is used in this work is a pioneering technology that demonstrates the practical feasibility of CCT with a broadband PHY based on Orthogonal Frequency-Division Multiplexing (OFDM). An evaluation of the dURLLC system in testbed experiments shows that it can meet the URLLC requirement of 5G networks, i.e., it can deliver a 32-byte MAC service data unit (MSDU) with a reception probability of 99.999 % at PHY data rates of up to 48 Mbit/s and at a latency per hop of down to 48.2 µs [11]. While the dURLLC system is based on the IEEE 802.11 OFDM PHY, dURLLC technologies could potentially also be integrated into future 6G networks.

Figure 3 illustrates the operation of the dURLLC system in the teleoperation testbed, showing two consecutive steps of a data packet transfer. In Figure 3a, the robot on the left initiates the data packet transfer by transmitting a data frame. This is the primary transmission, which can be received by both the robot on the right and by the three relay nodes of the dURLLC system. In Figure 3b, the three relay nodes retransmit the data frame as a CCT, immediately after their reception of the primary transmission. The CCT is the secondary transmission



(a) Primary transmission

(b) Secondary transmission as CCT

Fig. 3: Data transfer in two steps by the dURLLC system

and serves for the right robot as an additional reception opportunity of the same data. Thus, if the right robot receives the primary transmission successfully, it ignores the secondary transmission. But if it fails to receive the primary transmission, it uses the secondary transmission instead. Further, since the secondary transmission is a CCT, the relaying system itself is robust due to multiple reception opportunities at the relay nodes. If at least one of the relay nodes receives the primary transmission, the secondary transmission is generated. And since the secondary transmission is generated as a CCT, all the relay nodes can start transmitting without additional waiting or coordination. While Figure 3 illustrates the transfer of a data packet in one direction, the teleoperation testbed actually uses two arrays of three relay nodes that operate on two different frequency channels for bidirectional data transfers.

Each relay node consists of a Wireless Open-Access Research Platform (WARP) v3 Software-Defined Radio (SDR) [15] running a custom design [11, Chapter 4] that is based on the IEEE 802.11 reference design for WARP v3 [16]. The custom design essentially comprises signal processing algorithms for precise frequency synchronization running on the Field Programmable Gate Array (FPGA) of the WARP v3 and a medium access control (MAC) protocol for CCT-based communications [11, Chapter 4]. The MAC protocol allows the initiator of a data transfer to configure the retransmission behavior of the relay nodes and also comprises a sequence number that allows receivers to identify new and duplicate data frame receptions. The time synchronization among relay nodes is accomplished through hardware timers, posing several hard real-time deadlines. Each retransmission of a data frame starts 16.2 µs after the preceding reception, with only small deviations in case of propagation delay differences. With this, a 32-byte MSDU is transferred with a deterministic latency per hop of 48.2 µs by the relay nodes [11, Chapter 4].

To integrate the dURLLC system into the testbed, the robot computers are equipped with Atheros AR9285 IEEE 802.11 interfaces, which support operation in monitor mode with the frame injection capability of libpcap [17]. On the transmitter side, a custom bridging application (dURLLC bridge) reads the UDP data stream containing the packets for transmission from a raw socket and transfers them to the IEEE 802.11 interface. In this process, the application encapsulates the

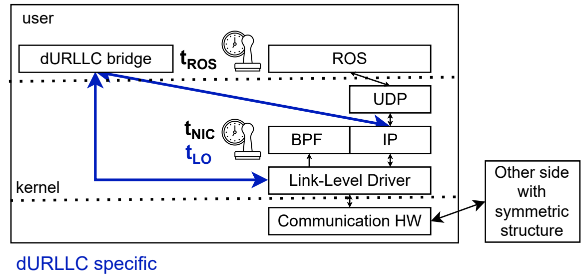


Fig. 4: TCP/IP layer overview for timestamp (t) locations

payload of each UDP packet into a data frame complying with the MAC protocol of the dURLLC system and sets the parameters appropriately. Then, it injects the data frame for immediate transmission on the IEEE 802.11 interface by means of libpcap. On the receiver side, another custom bridging application listens on a second IEEE 802.11 interface for data frames transmitted from the respective other robot and the corresponding dURLLC relay nodes. This application uses a Berkeley Packet Filter (BPF) to filter for the data frames of interest and discards duplicate data frame receptions. It further extracts the MSDU from each new data frame and generates a UDP packet. Each such packet is transferred to a raw UDP socket to which ROS 2 listens.

V. Experimental Methodology

This section describes the methodology. After timestamps for latency measurements are characterized, key metrics for evaluating the platform and subsystem performance are defined.

A. Timestamps

To evaluate the timing of the tested communication systems, timestamps are recorded in nanoseconds-precision Unix time at critical points in the data flow, as illustrated in Figure 4. For the evaluation, only one ROS 2 topic for each direction was used, as the teleoperation system sends a single state update with all topics per cycle. Specifically, the topic *joint_states* (124 bytes per message) was logged from the leader to the follower, while *tau_ext_hat* (48 bytes per message) was used from the follower to the leader. These topics contain the longest messages transmitted in the system, posing the greatest challenge for wireless communication and thus providing a robust basis for performance evaluation. For instance, timestamps $t_{\text{ROS\_TX}}$ sender-side and $t_{\text{ROS\_RX}}$ receiver-side capture the data flow from the transmitting ROS 2 node to the receiving ROS 2 node, including transitions through the network stack and link-level drivers. For the dURLLC system, the timestamps t_{NIC} and t_{LO} allow measuring the timing of the dURLLC bridge. The timestamps are defined as follows.

i) $t_{\text{ROS\_TX}}$ (sender-side): $t_{\text{ROS\_TX}}$ is set during creation of the message header in the respective node of leader or follower. This timestamp acts as a reference point for latency calculations and as a unique identifier for a message without adding additional network load.

ii) t_{NIC}: Timestamp t_{NIC} is derived from a tcpdump capture of messages from the Real-Time Publish-Subscribe (RTPS) protocol used by DDS for message delivery. This timestamp is recorded right before passing a message to a NIC for transmission and when receiving a message from it. For data analysis, RTPS messages are dissected to extract the timestamp t_{ROS} and the message type as identifiers. Additionally, the corresponding BPF capture timestamps are assigned as $t_{\text{NIC\_TX}}$ (sender-side) or $t_{\text{NIC\_RX}}$ (receiver-side).

iii) $t_{\text{ROS\_RX}}$ (receiver-side): $t_{\text{ROS\_RX}}$ is recorded when the ROS 2 subscriber callback function is executed, marking the completion of message processing in the application layer.

iv) t_{LO} and t_{NIC} for dURLLC system: The timestamp t_{LO} is captured when ROS 2-converted UDP messages arrive at the localhost interface listened to by the dURLLC bridge application. The timestamp t_{NIC} is logged right before a message is bridged onto the IEEE 802.11 interface for transmission. The duration between t_{LO} and t_{NIC} is the processing time of the dURLLC bridge application, for both sides.

To verify the accuracy of the BPF timestamps, the sum of all partial latencies is compared to the overall $t_{\text{ROS\_TX}}$ to $t_{\text{ROS\_RX}}$ (ROS-ROS) latency. This comparison demonstrates that the sum of partial latencies equals the total ROS-ROS latency with nanosecond accuracy, confirming the precision of the BPF timestamping method.

B. Performance Metrics

i) Latency: Latency is calculated as the time difference between two timestamps. For example, the end-to-end latency between ROS 2 nodes is computed as:

$$L_{\text{ROS-ROS}} = t_{\text{ROS\_RX}} - t_{\text{ROS\_TX}}. \tag{1}$$

The latency measurements are labeled according to their timestamp locations, with the naming convention *source-destination* (e.g., `ROS-NIC` or `NIC-LO`). A lost message is assigned a value of infinity.

ii) Latency Bound Probability (LBP): LBP is defined as the probability that a given latency can be maintained within a specified threshold, serving as a quantitative measure of system reliability. It is expressed as

$$LBP = P(L \leq L_{\text{threshold}}), \tag{2}$$

where $L_{\text{threshold}}$ is the maximum acceptable latency for a given application. Empirical Cumulative Distribution Functions (ECDFs) are subsequently used to visualize the fractions of packets meeting a latency threshold.

iii) Packet Delay Variation (PDV): PDV, also referred to as jitter, is calculated as the mean of the absolute latency differences between consecutive packets [18]:

$$PDV = \frac{1}{N-1} \sum_{i=1}^{N-1} |L_{i+1} - L_i|, \tag{3}$$

where N is the total number of packets and L_i is the latency of the i-th packet.

iv) Packet Loss: A lost packet is identified when a packet with the corresponding $t_{\text{ROS\_TX}}$ timestamp is missing from the measurement data of subsequent timestamps, indicating that it was not successfully received or processed at this stage.

VI. Results and Discussion

This section examines the performance of different communication technologies integrated into the testbed. All measurements reflect bidirectional communication and are based on datasets of over 200,000 packets per configuration. The maximum PTP time offset between the clocks of the leader and the follower computer was measured as 416 ns, which is negligible for the measurement results presented here. For single 5G tests, RM520N modems are used. The 5G-dual configuration utilizes both modems simultaneously on the same 5G system. The analysis covers the following key aspects: overhead of the teleoperation system, end-to-end performance including latency characteristics, reliability, and packet delay variation across technologies.

A. Teleoperation System

The ROS-NIC (sending) and NIC-ROS (receiving) latencies in Table I represent the overhead of the networking stack and ROS 2 implementation for sending and receiving data. The Inter-Departure Time (IDT) statistics (Mean = 1.002 ms, σ = 0.09 ms) confirm the system's capability to fully utilize the robot's 1000 Hz data interface, providing a stable foundation for evaluating various communication technologies.

TABLE I: Teleoperation specific latencies and IDT in ms, measured using Ethernet data

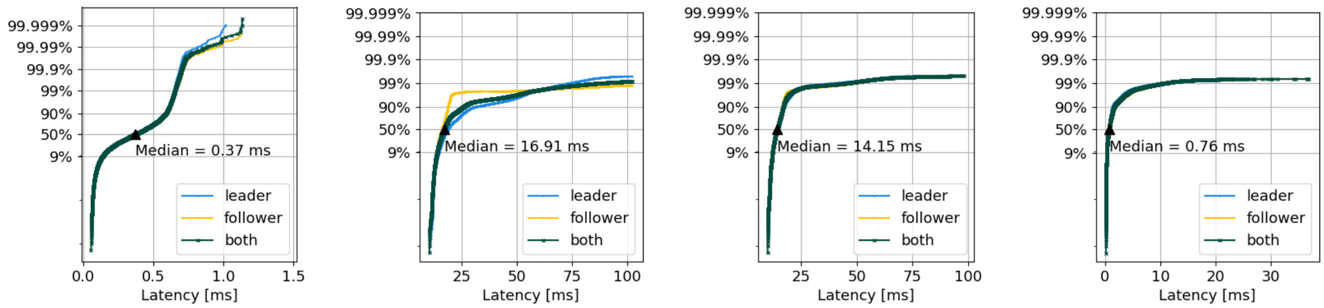
$L_{\text{ROS-NIC}}$	Median	0.3	IDT Mean	1.002
$L_{\text{ROS-NIC}}$	Max	1.45	IDT σ	0.09
$L_{\text{NIC-ROS}}$	Median	0.04		
$L_{\text{NIC-ROS}}$	Max	0.73		

B. Latency and Reliability

This section presents $L_{\text{NIC-NIC}}$ (communication system) and $L_{\text{ROS-ROS}}$ (end-to-end) performance metrics of the four integrated communication systems. Note that, as anticipated, Ethernet performs best and thus serves as a performance baseline.

Table II shows the NIC-NIC latency in milliseconds and covers the overall median value as well as the latency bounds under which certain reception probabilities can be reached. It shows that the dURLLC system encounters a high median latency of $\sim$100 ms when the teleoperation system is operated at an update rate of 1000 Hz, whereas it provides a low median latency of 0.14 ms at a sampling rate of 200 Hz. Since the dURLLC system itself offers a deterministic timing under any load [11], the performance degradation at a sampling rate of 1000 Hz can be attributed to the IEEE 802.11 interfaces used for packet injection and reception. These interfaces are operated without driver modifications, so buffering and exponential back-off mechanisms may in fact introduce delays under high load.

Further, the dURLLC system offers a reliability of about 99 %, but does not reach 99.9 %, which can be explained as follows: (1) Commercial IEEE 802.11 interfaces are used for the integration, which have a worse reception performance than SDRs with a soft-decision Viterbi decoder [11]. (2) To



(a) Ethernet (b) 5G (c) 5G-dual (d) dURLLC @ 200 Hz

Fig. 5: Latency $L_{\text{ROS-ROS}}$ of the four integrated communication systems

TABLE II: Latencies $L_{\text{NIC-NIC}}$ in ms

	Median	99 %	99.9 %	99.99 %	99.999 %
Ethernet	0.02	0.03	0.04	0.04	0.05
5G	16.14	89.22	-	-	-
dURLLC@1000 Hz	100.87	-	-	-	-
dURLLC@200 Hz	0.14	11.1	-	-	-

optimize latency, the dURLLC system is operated at the highest data rate of the IEEE 802.11 OFDM PHY, i.e., 54 Mbit/s, which comes at the cost of a reduced frame reception rate [11].

Table III shows the ROS-ROS packet loss rates of the four integrated communication systems. The single 5G system has a packet loss rate of 0.9 %, whereas the dual 5G system has a packet loss rate of 0.52 %, indicating an improvement of reliability by multi-connectivity.

TABLE III: End-to-end packet loss between ROS 2 nodes

	Eth.	5G	5G-dual	dURLLC@200 Hz
Num. of Packets	206,287	206,170	206,283	201,530
Abs. [ms]	0	1,853	1,068	1,386
%	0	0.9	0.52	0.69

Figure 5 shows the ROS-ROS ECDFs of the fraction of received packets over the latency for the four integrated communication systems. These plots essentially visualize the respective LBP metrics. Note that the dual 5G system has a slightly lower median latency than the single 5G system.

C. Packet Delay Variation (Jitter)

The PDV data in Table IV reveals that Ethernet's jitter primarily originates from the software stack, with a significant increase from NIC-NIC (0.002 ms) to ROS-ROS (0.2 ms). The 5G system's inherent PDV of 1.68 ms at NIC-NIC level slightly improves at ROS-ROS through dual connectivity, with a reduction of ca. 11 % from 1.52 ms to 1.36 ms. For dURLLC at 200 Hz, the software stack similarly influences timing, with ROS-ROS PDV of 0.56 ms exceeding NIC-NIC with 0.49 ms. The table also shows each PDV in relation to its median latency.

VII. Conclusion and Outlook

This work introduces a robotic teleoperation platform integrating multi-connectivity capabilities, enabling seamless

TABLE IV: Packet Delay Variation (Jitter)

	Eth.	5G	5G-dual	dURLLC @200 Hz
ROS-ROS abs. [ms]	0.2	1.52	1.36	0.56
ROS-ROS % of Median $L_{\text{ROS-ROS}}$	54.52	8.96	9.4	73.75
NIC-NIC abs. [ms]	0.002	1.68	-	0.49
NIC-NIC % of Median $L_{\text{NIC-NIC}}$	7.72	10.4	-	339.67

integration of wireless communication solutions. The system's performance is validated using Ethernet as a baseline, demonstrating a mean inter-departure time of packets at ROS 2 level of 1.002 ms. This software performance enables the teleoperation system to fully utilize the robot's 1 kHz cycle frequency, establishing a robust foundation for assessing a wide range of communication technologies.

The results indicate that a 5G system with dual connectivity improves the performance in comparison to 5G with single connectivity. The dual 5G system reduces the application level latency by approximately 16 % and the Packet Delay Variation (PDV) by 11 % compared to the single 5G system.

A decentralized Ultra-Reliable Low-Latency Communication (dURLLC) system was successfully integrated and offers performance comparable to 5G in terms of reliability at an update rate of the teleoperation system of 200 Hz. However, the IEEE 802.11 integration interfaces impose performance limitations, especially at higher cycle frequencies.

Future work will extend the platform to support video feedback and additional sensory data, such as depth information to enhance the operator's situational awareness. While video integration will increase bandwidth requirements and communication system load, these efforts will ensure the platform evolves to meet the demands of increasingly complex teleoperation scenarios.

Additionally, strategies for reducing network traffic in parallel redundancy setups will be explored to improve efficiency, alongside investigations into different physical configurations to assess their impact on the performance of the teleoperation.

Acknowledgment

This work has been supported by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 16KISK003K and 16KISK014, Open6GHub and 16KIS2239K, SUSTAINET-guarDian). The authors alone are responsible for the content of the paper.

REFERENCES

[1] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The Road Towards 6G: A Comprehensive Survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021, ISSN: 2644-125X. DOI: 10.1109/OJCOMS.2021.3057679.

[2] R. Das, N. J. Baishya, and B. Bhattacharya, "Areview on telemanipulators for remote diagnostic procedures and surgery," *CSI Transactions on ICT*, vol. 11, no. 1, pp. 31–37, Apr. 2023, ISSN: 2277-9078, 2277-9086. DOI: 10.1007/s40012-023-00373-2.

[3] F. Gosselin and M. Grossard, *Design and Development of a Dexterous Teleoperation Setup for Nuclear Waste Remote Manipulation*, O. Gusikhin, K. Madani, and H. Nijmeijer, Eds. Cham: Springer International Publishing, 2023, pp. 80–105, ISBN: 978-3-031-26474-0.

[4] K. Polachan, J. Pal, C. Singh, T. V. Prabhakar, and F. A. Kuipers, "TCPSbed: A Modular Testbed for Tactile Internet-Based Cyber-Physical Systems," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 796–811, Apr. 2022, ISSN: 1063-6692, 1558-2566. DOI: 10.1109/TNET.2021.3124767.

[5] I. Harjula, M. Uitto, M. Jurmu, J. Koskinen, J. Mäkelä, S. Walter, M. Hentula, T. Heikkilä, M. Lintala, and K. Rautiola, "Smart Manufacturing Multi-Site Testbed with 5G and Beyond Connectivity," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 1–6. DOI: 10.1109/PIMRC50174.2021.9569284.

[6] T. Farnham, S. Jones, A. Aijaz, Y. Jin, I. Mavromatis, U. Raza, A. Portelli, A. Stanoev, and M. Sooriyabandara, "UMBRELLA Collaborative Robotics Testbed and IoT Platform," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1–7. DOI: 10.1109/CCNC49032.2021.9369615.

[7] J.-B. Castillo-Sánchez, E. González-Parada, and J.-M. Cano-García, "Anovel testbed for evaluating ROS 2 robot swarm wireless communications," in *2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON)*, 2024, pp. 68–73. DOI: 10.1109/MELECON56669.2024.10608676.

[8] S. Ayvaşık, E. Babaians, A. Papa, Y. Deshpande, A. Jano, W. Kellerer, and E. Steinbach, "Demo: Remote Robot Control with Haptic Feedback over the Munich 5G Research Hub Testbed," in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2023, pp. 349–351. DOI: 10.1109/WoWMoM57956.2023.00059.

[9] J. Petershans, J. Herbst, M. Rüb, E. Mittag, and H. D. Schotten, "Robotic Teleoperation: A Real-World Test Environment for 6G Communications," in *28. ITG-Fachtagung Mobilkommunikation*, VDE-Verlag, May 2024, pp. 106–111.

[10] Franka Robotics, *franka_ros*, https://github.com/frankaemika/franka_ros, Accessed: 2025-01-20, 2025.

[11] R. Klose, "Decentralized Ultra-Reliable Low-Latency Communications through Concurrent Cooperative Transmission," Ph.D. dissertation, Darmstadt, Jul. 2023. DOI: 10.26083/tuprints-00024070.

[12] M. Zimmerling, L. Mottola, and S. Santini, "Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–39, Dec. 2020, ISSN: 0360-0300. DOI: 10.1145/3410159.

[13] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, *Efficient Network Flooding and Time Synchronization with Glossy*. Chicago, Illinois, USA: IEEE, Apr. 2011, pp. 73–84, ISBN: 978-1-4503-0512-9.

[14] B. A. Nahas, A. Escobar-Molero, J. Klaue, S. Duquennoy, and O. Landsiedel, "BlueFlood: Concurrent Transmissions for Multi-hop Bluetooth 5 – Modeling and Evaluation," *ACM Transactions on Internet Things*, vol. 2, no. 4, pp. 1–30, Jul. 2021. DOI: 10.1145/3462755.

[15] *Wireless Open Access Research Platform*, http://warpproject.org, Accessed: 2025-01-17.

[16] *Wireless Open Access Research Platform: 802.11 Reference Design for WARP v3*, https://warpproject.org/trac/wiki/802.11, Accessed: 2025-01-17.

[17] *Website of tcpdump, a powerful command-line packet analyzer, and libpcap, a portable C/C++ library for network traffic capture*, https://www.tcpdump.org, Accessed: 2025-01-17.

[18] J. Kurose and K. Ross, *Computer Networks: A Top-Down Approach Featuring the Internet*. Pearson Addison Wesley, 2010.