Image and Command Transmission Over the 5G Network for Teleoperation of Mobile Robots

Thiago B. Levin[1,2], João Miguel Oliveira[1,2], Ricardo B. Sousa[1,2], Manuel F. Silva[2,3], Bruno S. Parreira[4], Héber Miguel Sobreira[1], and Hélio Sousa Mendonça[1,2]
[1]CRIIS – Centre for Robotics in Industry and Intelligent Systems, INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. Portugal. {thiago.levin,joao.c.oliveira, heber.m.sobreira}@inesctec.pt
[2]Faculty of Engineering, University of Porto (FEUP). Portugal. up201503004@edu.fe.up.pt, hsm@fe.up.pt
[3]ISEP, Polytechnic of Porto. Portugal. mss@isep.ipp.pt
[4]NOS SGPS. Portugal. bruno.parreira@nos.pt

***Abstract*—Human oversight can benefit scenarios with complex tasks, such as pallet docking and loading and unloading containers, beyond the current capabilities of autonomous systems without any failures. Furthermore, teleoperation systems allow remote control of mobile ground robots, especially with the surge of 5G technology that promises reliable and low latency communication. Current works research on exploring the latest features from the 5G standard, including ultra-Reliable Low-Latency Communication (uRLLC) and network slicing. However, these features may not be available depending on the Internet Service Provider (ISP) and communication devices. Thus, this work proposes a network architecture for the teleoperation of ground mobile robots in industrial environments using commercially available devices over the 5G Non-Standalone (NSA) standard. Experimental results include an evaluation of the network and End-to-End (E2E) latency of the proposed system. The results show that the proposed architecture enables teleoperation, achieving an average E2E latency of 347.19 ms.**

***Index Terms*—5G mobile network, remotely operated vehicles, teleoperation, mobile robots**

I. Introduction

Logistics and retail companies, warehouses, automotive and aerospace industries, agriculture businesses, and healthcare institutions use Autonomous Mobile Robots (AMRs) to automate the internal flow of materials, delivery processes, and security monitoring tasks. Automating these tasks instead of relying on human labour improves operational efficiency, flexibility, and productivity, achieving higher efficiencies and lower production costs [1]. Despite the progress in autonomous systems, there are scenarios where human expertise is indispensable. More specifically, human oversight and intervention can benefit scenarios with complex tasks which may be beyond the current capabilities of autonomous systems for automated execution without any fails, such as pallet docking, containers and trucks loading and unloading, among others.

Consequently, there is a need for teleoperation and remote control of mobile ground robots in industrial environments. Teleoperation systems require control mechanisms to enable the remote operation of autonomous vehicles. These mechanisms are negatively impacted by time-varying control input delays and intermittent communication. Indeed, latency and its variability (jitter) significantly influence the immersive experience of the operator [2]. With the launch and adoption of the 5G network by the ISPs, this network should offer improved data transmission speed with the enhanced Mobile BroadBand (eMBB) core, support of uRLLC, and allow high-density of devices connection with the Massive Machine-Type Communications (mMTC) core [3].

This paper proposes a 5G network architecture with commercially available communication devices to achieve the teleoperation of ground mobile robots in industrial environments. The system architecture includes using cameras to provide visual feedback to the operator via a video stream, implementing both the Real Time Streaming Protocol (RTSP) and Web Real-Time Communication (WebRTC), Message Queueing Telemetry Transport (MQTT) messages to handle the direct teleoperation control of the robot's linear and angular velocities, and the 5G Non-Standalone (NSA) network topology definition. Experimental results focus on the evaluation of the latency and jitter of RTSP and WebRTC video streaming protocols, with the proposed architecture achieving an average E2E latency of 347.19 ms. The results show that the architecture is capable of performing teleoperation through the 5G NSA.

The remainder of this paper is organised as follows. Section II reviews the literature on teleoperation of ground mobile robots and network mediums. Next, Section III presents the proposed 5G network architecture and control scheme for the teleoperation of an AMR in an industrial environment. Section IV analyses the experimental results obtained with the proposed system, including latency measurements. Finally, Section V formulates the final conclusions and future work.

II. Related Work

A teleoperation system allows the human operator to interact with an environment remotely. This interaction requires a link between the operator and the robot to connect them and enable communication and exchange of information.

This work is co-financed by Component 5 – Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021–2026, within project PRODUTECH R3, with reference 60.

2024 7th Iberian Robotics Conference (ROBOT) | 979-8-3503-7636-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ROBOT61475.2024.10797434

A. Bluetooth communication

Bluetooth is a possible communication medium for creating the link, benefiting from a low implementation cost and quick setup [4]. Nádvorník and Smutný [5] use Bluetooth due to its low power requirements with up 720 kbps data rate in asynchronous mode to perform direct teleoperation control of the robot's motor speed. At the same time, the operator has feedback in terms of obstacles from ultrasonic sensors. Similarly, Butnaru *et. al.* [6] control a robot via Bluetooth with feedback from proximity sensors. Both works [5] and [6] use peer-to-peer communication between robot and operator.

In contrast to [5] and [6], Xue *et. al.* [7] use a server to communicate with the operator's computer and the robot. While the communication server–robot is on a Bluetooth connection that supports up to 100 kbps, the server–teleoperator link uses a standard Ethernet local network. The authors also focuse on direct teleoperation but with visual feedback from a 352×288 px Complementary Metal Oxide Semiconductor (CMOS) colour camera, among other onboard sensors, including ultrasonic and infrared sensors. However, Bluetooth wireless connections suffer low range and bandwidth and are severely affected by obstacles in the scene [4].

B. Wireless Local Area Networks (WLAN)

Furthermore, another common topology is Wireless Local Area Networks (WLAN). Wang and Liu [8] propose a teleoperation system with supervisory control (linguistic and reference velocity commands). A 2.54 GHz WLAN interconnects the proprioceptive and exteroceptive sensory data from the robot and the camera feed sent to a streaming server to the Internet. Then, teleoperators consume the data and interact with the robot. However, the Internet telephone line limited the client's connection to 33.6 kbps. Similarly, both Wang *et. al.* [9] and Song *et. al.* [10] implement a WLAN also connected to the Internet to achieve remote control from an external location. While work [9] allows both direct and supervisory teleoperation (travel distance or to a specific point) from sensor information, robot position, and live video data transmitted over the User Datagram Protocol (UDP), in [10] only focus on direct teleoperation with a shared control (robot reduces or inhibits the user's direct control commands to prevent any possible collision).

Caccamo *et. al.* [11] only use a WLAN with IEEE 802.11n 2.4 GHz channels to remotely control a ground mobile robot through direct teleoperation in an Search And Rescue (SAR) application. The authors concluded that providing network connectivity information is helpful for the intended SAR application. Moreover, Birk *et. al.* [12] underline the importance of data serialisation and video compression with motion Joint Photographic Experts Group (JPEG) format instead of video codecs to achieve lower latency in wireless networks. Finally, Tikanmäki *et. al.* [13] differ from the previous works by adopting the WLAN IEEE 802.11ac 5 GHz standard, achieving up to 1000 Mbits/s at close range and almost 400 Mbit/s in the 30 m range. This use case includes the WLAN connection between a ground robot and a quadcopter (DJI Inspire 1) using a bridge configuration of two WLANs to provide fast, multichannel and long-distance communication. Also, [13] employs raw point cloud streaming and adaptive data reduction to reduce latency and increase the frame rate.

C. Mobile networks

Mobile 3G/4G networks may offer more extensive coverage, range, capacity, security, and possibly higher bandwidths, compared to WLAN networks [4]. Jincun *et. al.* [14] and Duong *et. al.* [15] employ 3G for their proposed teleoperation systems, while Uddin *et. al.* [16] adopt the 4G network. Work [14] proposes a long-range remote control system for rescue robots based on 3G, where both the supervisory and control centre and the robot are connected to the Internet. Moreover, [15] proposes a teleoperation system with both direct and supervisory control, where the client is connected directly to the Internet and the robot to a 3G network using a USB device. The system implements the video, sensing data, and administrative/supervisory control data transmission over Real-time Transport Protocol (RTP), UDP, and Transmission Control Protocol (TCP), respectively. Results show a 103 ms time delay of the network for 100 B and 255 ms for 2000 B, considering a distance of 30 km. As for Uddin *et. al.* [16], the teleoperation system connects the robot to the Internet through a 4G modem for supervisory teleoperation (translation and rotation motions). However, the proposed system does not provide any feedback from the environment.

Moreover, Shen *et. al.* [17] propose a Virtual Reality (VR)-based teleoperation system, using the RTP to transmit the video feed over the GStreamer pipeline, and the other control signals utilise the Robot Operation System (ROS). The experimental results compare WLAN, 3G, and 4G networks. Although 3G/4G mobile networks offer more range and broader area coverage than WLAN, the 3G/4G available bandwidth may depend on proximity to network towers, number of users, and vehicle speed. Also, the WLAN network offered the best teleoperation experience and lower network latency (average of 106.5 ms for video, with a standard deviation of 10.5 ms, compared to 183.9 ms and 16.7 ms in 4G, respectively).

With the 5G development, this network is designed to enhance the experience of mobile devices, including the Internet of Things (IoT), mobile vehicles (V2X), and eMMB experience, while supporting uRLLC equipment for remote applications and industrial automation [3]. Moreover, 5G networks operate on other frequency ranges designated explicitly for them. While WLAN is widely used in industrial environments nowadays, these networks cannot be used for communication requiring high reliability, since most devices in the industries are already networked via free Industrial, Scientific, and Medical (ISM) bands [18]. Thus, European-funded projects, such as 5G-TOURS [19], 5G-Blueprint [20], 5G-HEART [21], and 5G-ERA [22], demonstrate future use cases for employing 5G networks, including teleoperation. These projects research on improving the throughput by aggregating different carriers of the same or different technologies, such as 4G and 5G, in the case of 5G NSA [19], network slicing, Multi-access Edge

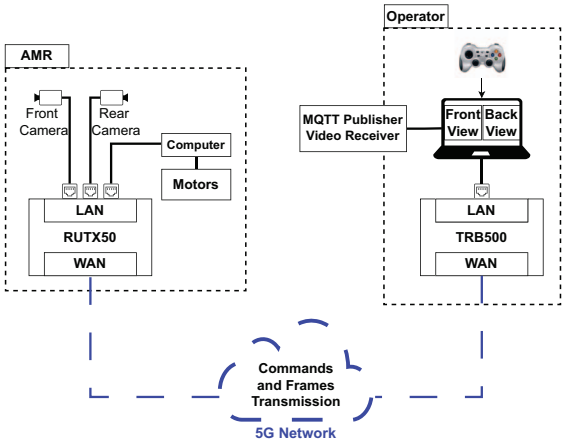


Fig. 1: Architecture of the teleoperation system proposed in this work over a 5G Non-Standalone (NSA) network.

Computing (MEC), Network Function Virtualization (NVF), Service-Based Architecture (SBA) [20]. Moreover, the research projects validate the usage of 5G networks in health, transport, and aquaculture environments [21], and cloud-native network applications in order to enhance robot autonomy and supports continuous deployment and integration through a cloud-native application approach [22]. Nevertheless, the experience of teleoperation applications with commercial 5G networks depends on the evolution and adoption of the 5G standard by the ISPs and the communication devices.

III. Teleoperation System Architecture

This work proposes a system architecture to enable teleoperation through the 5G network with commercial communication devices. The proposed teleoperation system, illustrated in Figure 1, consists of two cameras on the AMR to provide visual feedback to the operator from the robot's surroundings, where one is for the front view and the other for the rear view. Furthermore, the operator is in a remote location with a computer running the video player application for visual feedback from the cameras and the command transmission application, in which the operator interacts with a joystick to perform direct teleoperation over the velocity commands of the robot. All the communications are over a 5G-based network architecture, set up at the iiLab – industry and innovation Lab[1] facilities from INESC TEC, powered by the NOS SGPS ISP.

A. 5G Non-Standalone (NSA) mobile network

First, the proposed teleoperation system operates in the network infrastructure at the iiLab facilities and emulates an industrial environment. This network infrastructure implements a Radio Access Network (RAN) consisting of the 5G NSA standard following the 3GPP R15 release, with different carriers for the 4G Long-Term Evolution (LTE) and 5G technologies (in this case, 2.6 GHz and 3.5 GHz, respectively). Still, the network infrastructure is based on the 4G LTE technology because the ISP does not yet support the 5G Standalone (SA) standard.

Furthermore, the RAN is employed in the industrial environment by small cells from the Ericsson Radio DOT system: Ericsson DOTs B78L 3.5 GHz 4×4 Multiple Inputs Multiple Outputs (MIMO) for the 5G carrier band, and Ericsson DOTs B7 2.6 GHz 2×2 MIMO for the 4G band. The Indoor Radio Units (IRUs) power and control the Radio DOTs through the IRU 2242 and IRU 8846 devices for the 4G LTE anchoring system and the 5G technology, respectively. Then, the Baseband 6630 is connected to the IRUs to provide the pooled baseband resources for the environment area. The connections between all these radio units is ensured by Ethernet CAT6a cables. This RAN setup employed in this work provides enhanced connectivity to the 5G core without being solely dependent on the public radio cells.

As illustrated in Figure 1, both the AMR and the teleoperator systems have 4G/5G compatible routers. The robot uses a Teltonika RUTX50 industrial router to connect to the 5G NSA network through a Subscriber Identity Module (SIM) card provided by the ISP to enable the connection to the dedicated 5G NSA RAN. This router has four RJ-45 ports, connecting the two cameras used in this work via Ethernet cables to provide visual feedback to the operation. In the case of the teleoperator system, a Teltonika TRB500 5G Gateway with a SIM card connects the computer to the network. Also, in order to enable access to the cameras and MQTT topics outside of AMR and the teleoperator system's Local Area Network (LANs), the ports 554 (camera feed) and 1883 (MQTT communication) are forward to three external ports, two for the two cameras and one for the computer.

B. Command transmission

The MQTT protocol[2] is chosen for the proposed teleoperation architecture. This choice stems from the fact that the AMR used in the experiments employs the MQTT protocol as the bridge between external connections and its internal software system. Nevertheless, MQTT is widely used for machine-to-machine communication in industrial scenarios. This protocol implements a lightweight publish-subscribe messaging pattern, focusing on low bandwidth usage. Moreover, MQTT uses a broker with the ability to handle multiple clients, facilitating the configuration of new devices, functionalities, and enabling the scalability of the teleoperation system. The protocol only supports TCP communication, guaranteeing delivery within the same network. Finally, MQTT has support for basic Quality of Service (QoS) functionality, user authentication, and data encryption [23].

Consequently, the remote operator and the AMR represent two clients in the data communication structure for this study, illustrated in Figure 2. The operator client publishes the commands in the */cmd_vel* MQTT topic in JavaScript Object Notation (JSON) format, which contains information

[1]https://www.inesctec.pt/en/laboratories/iilab-industry-and-innovation-lab

[2]https://mqtt.org/

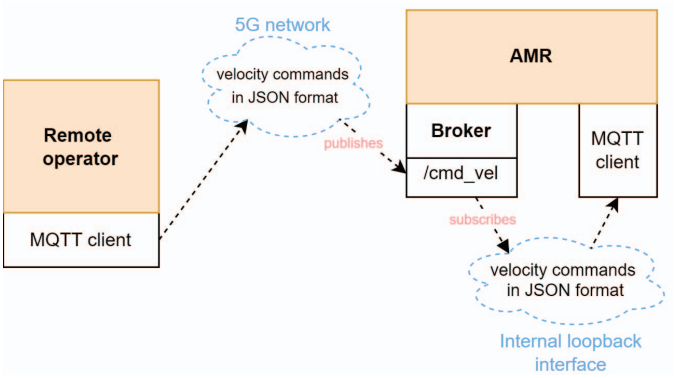


Fig. 2: MQTT data communication structure.

on linear ("v" for forward/backward, "vn" for left/right) and angular ("wn") reference velocities for the robot. This information makes the commands generic for any steering geometry of the teleoperated vehicle, independently of its holonomic constraints. Then, the vehicle client subscribes to the */cmd_vel* to set the desired reference velocity for the robot's internal software. In this study, the broker runs on the vehicle's computer to reduce latency, allowing the vehicle's MQTT client to publish and subscribe information through an internal loopback interface, instead of running it through the network. However, in a multi-vehicle teleoperation application, the broker may run on a separate server to offload the broker computational resources from the robots. Additional topics would also have to be introduced for each individual robot.

C. Video Streaming

The video streaming system captures, encodes, and transmits video from one node to another over a network. The streaming architecture can comprise two or three nodes, depending on the protocol. Moreover, three of the most common video streaming protocols are Real Time Messaging Protocol (RTMP), RTSP, and WebRTC. When using RTMP and RTSP, the video source and encoder on the sender side transmit the encoded video over the network to the video server, which then forwards it to the video player to decode and play the streamed video. In contrast, WebRTC allows peer-to-peer communication between the receiver and sender. Thus, the streaming transmission does not require an intermediate server. Overall, each component in the video streaming pipeline introduces a delay component, and the sum of these delays gives the E2E latency.

This work employs two low-cost Internet Protocol (IP) cameras Tapo C310 with an embedded H.264 encoder as video sources. The cameras support the RSTP protocol by default, while also being compatible with WebRTC. Indeed, the Tapo C310 camera captures the frames, compresses them, and sends them to the RTSP server, which the video player can access via the RTSP link with the defined user and password. Communication between server and player can take place either via TCP or UDP, even though only UDP transmission is considered in this study. Regarding technical specifications, the Tapo C310 has a frame rate of 15 fps and a 1920×1080 streaming resolution.

Therefore, the video player that processes the streaming data runs on the teleoperator side. The player receives the images, decodes, and displays them. As a result, this node introduces decoding and display delays. This work uses two video player applications, explained next, to compare the impact of using RSTP versus WebRTC as streaming protocols. Also, in order to start the video streaming, a request is made to the RUTX50 (see the network architecture in Figure 1) via an RTSP link (WebRTC also requires RTSP as a signalling server) that refers to the IP of the Wide Area Network (WAN) interface and the external port, defined on a forwarding configuration. This request is sent through the TRB500 gateway that forwards the request from the teleoperator side to the 5G network. Then, the request is received by the cameras located on the robot, connected to the 5G network through the RUTX50 router.

1) Real Time Streaming Protocol (RTSP): The RTSP [24] is an application-level protocol designed to transmit real-time streaming media. This protocol establishes and controls media sessions between endpoints, managing a session state using session identifiers. Servers using this protocol can be positioned at different points in the network and act as intermediaries between the media source and the client. Moreover, RTSP is usually based on the RTP and Real-time Transport Control Protocol (RTCP) protocols. A client can request a media stream from an RTSP server by sending an RTSP request to the server. Then, the server responds with the requested stream using RTP. RTSP is often used to control the streaming of multimedia content, such as video and audio.

A video player application compatible with RTSP, developed in the scope of this work is implemented in C++ using the OpenCV library[3]. This application requests access to the RTSP server, which then the latter streams the video over the network back to the video player. The player application decodes and displays the video using the ffmpeg library available in OpenCV with an H.264 decoder.

2) WebRTC protocol: WebRTC [25] is an open-source project that provides real-time communication for web browsers and applications via Application Programming Interfaces (APIs). This protocol supports video, voice, and general data transfers between peers. Unlike other streaming protocols like RTSP or RTMP, WebRTC is designed for peer-to-peer communication. As a result, the media streams are sent directly between clients without an intermediary server, reducing latency. Furthermore, WebRTC includes several key components, such as signalling server to initiate and manage the initial communication setup between peers, session traversal utilities to address Network Address Translation (NAT) and firewall traversal to achieve peer-to-peer communication, and secure audio and video transmission by using the Secure Real-Time Protocol (SRTP) to ensure that media streams are protected from eavesdropping and tampering.

[3]https://github.com/opencv/opencv

The *Ispy Agent DVR*[4] application is used as the video player in this work to test the WebRTC protocol and provide visual feedback to the teleoperator. This application is an open-source surveillance video application that uses WebRTC to connect the user to an IP camera. When using the *Ispy Agent DVR* application, the WebRTC protocol uses the RTSP server as a signalling server to establish peer-to-peer communication to the camera. Then, the image transmission is sent over the 5G NSA network to the teleoperator computer.

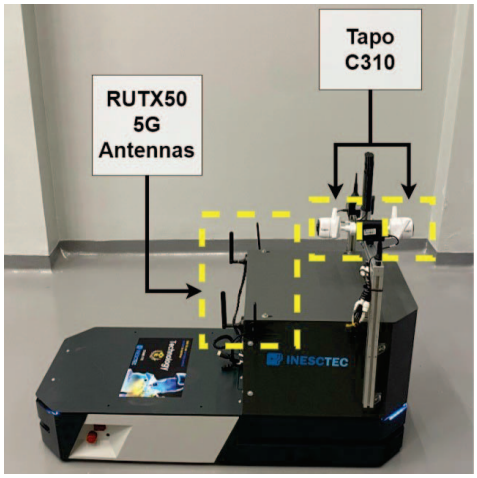
IV. Tests and Results

The experiments used the mobile ground robot Go-Mouse AMR400 from Flowbotic[5]. Figure 3 illustrates both the AMR and the visual feedback provided to the teleoperator from the two Tapo C310 cameras. In terms of the experimental methodology, first, the performance of three MQTT brokers (Mosquitto[6], EMQX[7], and VerneMQ[8]) is evaluated in terms of mean latency and its jitter. Next, similarly, the My TraceRoute (MTR) is used to evaluate average latency and jitter for the network infrastructure. Finally, the other evaluation presented in this paper is the E2E latency for the teleoperation system of an AMR proposed in this work.

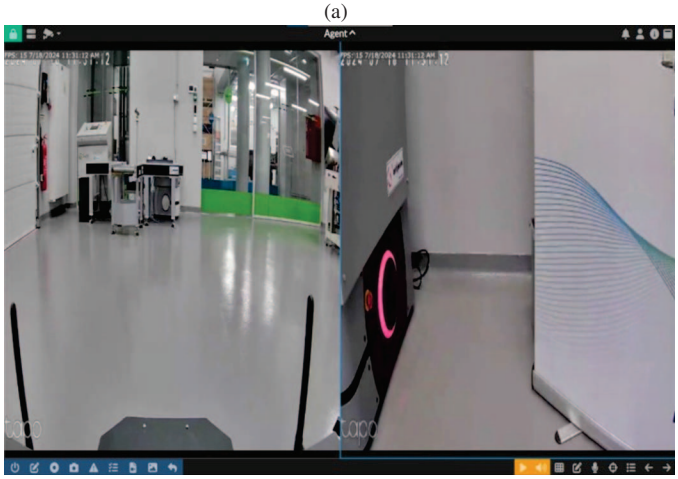
A. MQTT Broker Performance

In this work, three open-source implementations of an MQTT broker are considered for the teleoperation system: Mosquitto, EMQX, and VerneMQ. As a way to evalute the performance of those brokers, the *mqtt-benchmark*[9] tool is used to test a broker with a given number of simulated clients, message size, number of messages, and the QoS level. For the test, two clients are simulated representing the setup of one remote operator and one operated AMR with messages of 30 B, which originates from the size of the JSON command message previously presented in Section III. Furthermore, 100 messages are published to the broker, with an interval of one second between each. In order to understand the performance of each broker in different reliability levels, the tests are performed for each QoS level of the MQTT protocol. This test is done using a computer with the Ubuntu 20.04 Operating System (OS), where the three different brokers are implemented as services. For each test, only the service activated from the three brokers is the one being tested by the benchmarking tool. This setup relies solely on the internal loopback interface of the computer for the communication between the simulated clients to not have interference from the latency and possible jitter from the network infrastructure.

Table I presents the average latency and jitter results obtained from *mqtt-benchmark*. In terms of mean latency, the Mosquitto broker slightly outperforms both EMQX and VerneMQ, independently of the QoS. As for jitter, VerneMQ shows the best results, showing smaller values in this metric. Consequently, Mosquitto is chosen in this work as an MQTT broker for the teleoperation system. This choice is mainly due to a low average latency being seen as a primary factor for teleoperation applications. Similarly, the QoS level is set to level 0, given the lower latency obtained in the experiments.



(a)



(b)

Fig. 3: Experimental setup: (a) Flowbotic GoMouse AMR400; (b) visual feedback to the teleoperator.

TABLE I: Latency Metrics for Different Brokers and QoS Levels.

Broker	QoS	Average Latency (ms)	Jitter (ms)
Mosquitto	0	0.149	0.004
	1	0.428	0.022
	2	0.643	0.034
EMQX	0	0.151	0.016
	1	0.815	0.036
	2	1.095	0.003
VerneMQ	0	0.164	0.003
	1	0.688	0.005
	2	0.970	0.003

B. Network Latency

Next, the MTR tool available in Linux devices is used to test the average latency and jitter of the network infrastructure. MTR combines the functionalities of traceroute and ping into

[4]https://github.com/ispysoftware/iSpy?tab=readme-ov-file
[5]https://www.flowbotic.eu/
[6]https://mosquitto.org/
[7]https://www.emqx.com/
[8]https://vernemq.com/
[9]https://github.com/krylovsk/mqtt-benchmark

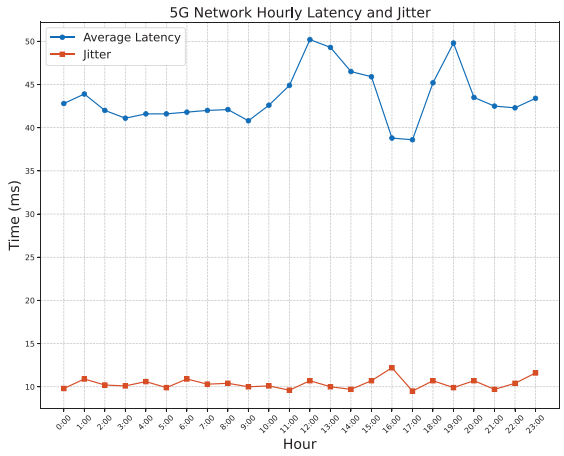


Fig. 4: Latency and jitter timing results throughout the day for the 5G Non-Standalone (NSA) network architecture proposed in this work.

a unified network diagnostic tool. Upon initialisation, MTR probes the network connection between the host and a specified hostname by transmitting packets with deliberately low Time-to-Live (TTL) values. The tool continues this process by monitoring the response times of intermediate routers. Then, MTR reports the percentage of responses received and their respective round-trip times along the internet route to the specified hostname.

A CronJob[10] is configured on the operator's computer to execute a MTR command hourly for 24h to gather data, sending 500 packets every hour to the RUTX50 where the AMR is connected to and saving the results report to a file. Figure 4 presents the average latency and jitter obtained from the CronJob execution. The average network latency between the robot and the operation station reveals a range from 38 ms to 50 ms throughout the day. Latency is relatively stable during the early morning hours, averaging around 40 ms. However, a significant increase occurs post–08:00, which coincides with typical working hours and suggests increased network congestion. As for the jitter results, this metric shows minimal fluctuation during the day, ranging from 9 ms to 10 ms. These results indicate stable packet delay and favourable network performance for teleoperation systems, given that the jitter may also be problematic for the teleoperation activity [2]. Overall, these latency and jitter results could be further improved with a 5G SA network infrastructure. Indeed, a recent study [26] presenting the main conclusions from the 5G-Blueprint project [20] refers that 5G SA plays an essential role to achieve the strict requirements for teleoperation. Nevertheless, the availability of the 5G SA is dependent on the ISP and the compatibility of the communication devices with the 5G SA specification.

[10]https://cron-job.org/

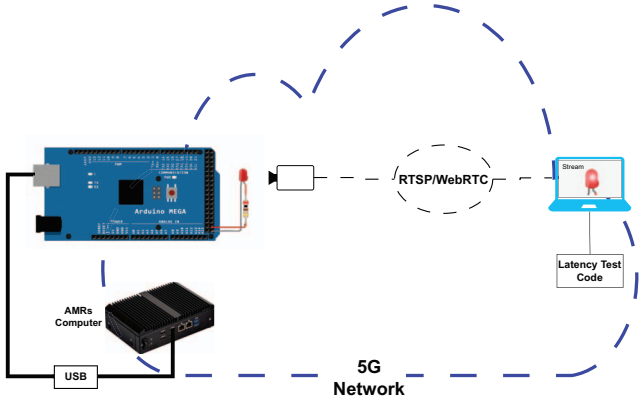


Fig. 5: Test methodology of the End-to-End (E2E) latency evaluation.

C. End-to-End (E2E) Latency

Lastly, Figure 5 illustrates the testing methodology considered in this paper to measure the E2E latency of the proposed teleoperation system. The teleoperator computer sends a boolean command in the JSON file, alongside the reference linear and angular velocities for the AMR, to turn a LED on and off. Then, the AMR computer processes the subscribed MQTT topic with the desired LED state and sends a command via a serial port with 115200 bps of baud rate to an Arduino Mega. The latter is the microcontroller that changes the physical state of the LED in terms of being on or off.

Furthermore, the LED is positioned within the field of view of the AMR's front camera to capture pixel colour alterations in the video transmission. Then, the OpenCV library is employed to analyse the computer screen for a colour change in a series of pixels, where the LED is located in the feedback image. A thread is implemented to ensure a 1.5 s delay before issuing the command to activate the LED, with the timestamp being saved as $t_{\text{cmd\_on}}$. If the pixel colour change is not detected within this period, a timeout occurs, the command is resent, and the $t_{\text{cmd\_on}}$ value is updated accordingly. Upon successfully recognition of the pixel colour alteration, the corresponding timestamp is saved as t_{LED}, and the command to deactivate the LED is dispatched by the teleoperator computer. As both timestamps are collected from the same system, the timestamp synchronisation is guaranteed, enabling the accurate calculation of total latency as formulated in Equation 1.

$$\Delta t_{\text{latency}} = t_{\text{LED}} - t_{\text{cmd\_on}} \tag{1}$$

Figure 6 provides the comparison of the E2E latency distribution results for the RTSP and WebRTC video players (see Section III), highlighting key differences in their performance characteristics. This comparison represents a data collection of at least 100 E2E latency results for each protocol using the network infrastructure proposed in this work. The median latency (indicated by the white dot within each violin plot) shows a significant difference between the two protocols. WebRTC has a median latency of 347.19 ms, reflecting relatively lower

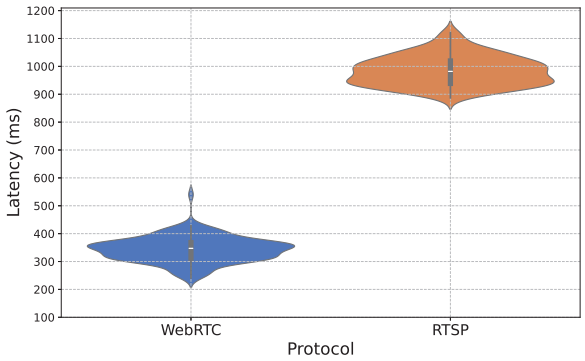


Fig. 6: End-to-End (E2E) latency distribution results for the teleoperation application using the WebRTC and RTSP streaming protocols. Median value represented by a white dot in the data distributions.

latency. The WebRTC latency distribution is also narrower and concentrated around the median, with a pronounced peak at 347.19 ms and limited spread. In contrast, RTSP exhibits a median latency of 981.98 ms, indicating substantially higher latency. The latency distribution for this protocol is broader and more dispersed when compared to the WebRTC results, indicating higher variability in latency.

Overall, the WebRTC protocol demonstrates a more stable and predictable behaviour in terms of latency, making it the preferred choice to develop a teleoperation application in comparison to RTSP. Also, the network latency represents around 12% of the total E2E latency, indicating that the video streaming application layer is the main cause for obtaining 347.19 ms of median E2E latency. As for the latency requirements for a teleoperation application, Moniruzzaman *et. al.* [2] refers that vehicle control while driving at ground speeds up to 90 km/h, equivalent to 25 m/s, degrades significantly once latency exceeds 170 ms. Although the proposed teleoperation system achieves a median latency of 347.19 ms, exceeding the 170 ms limit, this latency was considered acceptable from the teleoperator point-of-view. Indeed, the AMRs usually travel at around 1 to 2 m/s maximum linear velocity within industrial environments, being 12 to 25 times slower that the reference speed referred in [2].

V. Conclusions and Future Work

This study was able to successfully demonstrate the feasibility of teleoperation over a 5G NSA public network. The experiments included a comparative study of three MQTT brokers (Mosquitto, EMQX, and VerneMQ), network latency evaluation, and E2E latency analysis considering two different video streaming protocols (RTSP and WebRTC). As a result, the teleoperation system proposed in this work employs the Mosquitto MQTT broker and the WebRTC streaming protocol over a 5G NSA public network, achieving an average E2E latency of 347.19 ms in the remote control of an AMR in an industrial environment. This latency result was consider acceptable for industrial environments, considering the maximum linear velocity of 1 to 2 m/s for AMRs in those scenarios. As future work, improvements can be made on reducing both network and E2E latency. In terms of network infrastructure, the 5G NSA network may be upgraded to a 5G SA topology, while also exploring other capabilities of the 5G standard, such as uRLLC, network slicing, and the configuration of the SIM cards to ensure connectivity with the nearest 5G station. Regarding the streaming application layer, other cameras may be tested to evaluate their internal processing speed in terms of video streaming, while also developing an optimised application based on the WebRTC multimedia streaming protocol. Finally, the scalability of the MQTT protocol may be explored for teleoperation of multiple robots by introducing dedicated topics for each vehicle.

VI. Acknowledgements

The authors would like to thank NOS SGPS in the scope of the PRODUTECH R3 project for its support and assistance in developing this work.

References

[1] R. Raj and A. Kos. “A comprehensive study of mobile robot: history, developments, applications, and future research perspectives”. In: *Applied Sciences* 12.14 (2022), p. 6951. DOI: 10.3390/app12146951.

[2] M. Moniruzzaman, A.R., D. Chai, and S.M. Islam. “Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey”. In: *Robotics and Autonomous Systems* 150 (2022), p. 103973. DOI: 10.1016/j.robot.2021.103973.

[3] L. Qiao, Y. Li, D. Chen, S. Serikawa, M. Guizani, and Z. Lv. “A survey on 5G/6G, AI, and robotics”. In: *Computers and Electrical Engineering* 95 (2021), pp. 107–372. DOI: 10 . 1016 / j . compeleceng . 2021 . 107372.

[4] S. Opiyo, J. Zhou, E. Mwangi, W. Kai, and I. Sunusi. “A review on teleoperation of mobile ground robots: architecture and situation awareness”. In: *International Journal of Control, Automation and Systems* 19.3 (2021), pp. 1384–1407. DOI: 10.1007/s12555-019-0999-z.

[5] J. Nádvorník and P. Smutný. “Remote control robot using Android mobile device”. In: *2014 15th International Carpathian Control Conference (ICCC)*. 2014, pp. 373–378. DOI: 10 . 1109 / CarpathianCC . 2014 . 6843630.

[6] T. Butnaru, F. Gîrbacia, F. Tîrziu, and D. Talabă. “Mobile robot system controlled through mobile communications”. In: *Product Engineering: Eco-Design, Technologies and Green Energy*. Ed. by D. Talabă and T. Roche. Springer Netherlands, 2005, pp. 433–442. DOI: 10.1007/1-4020-2933-0_26.

[7] X. Xue, S.X. Yang, and M.Q.-H. Meng. “Remote sensing and teleoperation of a mobile robot via the Internet”. In: *2005 IEEE International Conference on Information Acquisition (ICIA)*. 2005, pp. 537–542. DOI: 10.1109/ICIA.2005.1635147.

[8] M. Wang and J.N.K. Liu. "Interactive control for Internet-based mobile robot teleoperation". In: *Robotics and Autonomous Systems* 52.2 (2005), pp. 160–179. DOI: 10.1016/j.robot.2005.04.001.

[9] D. Wang, J. Yi, D. Zhao, and G. Yang. "Teleoperation system of the Internet-based omnidirectional mobile robot with a mounted manipulator". In: *2007 International Conference on Mechatronics and Automation (ICMA)*. 2007, pp. 1799–1804. DOI: 10.1109/ICMA.2007.4303823.

[10] K.-T. Song, S.-Y. Jiang, and M.-H. Lin. "Interactive teleoperation of a mobile manipulator using a shared-control approach". In: *IEEE Transactions on Human-Machine Systems* 46.6 (2016), pp. 834–845. DOI: 10.1109/THMS.2016.2586760.

[11] S. Caccamo, R. Parasuraman, F. Båberg, and P. Ögren. "Extending a UGV teleoperation FLC interface with wireless network connectivity information". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 4305–4312. DOI: 10.1109/IROS.2015.7353987.

[12] A. Birk, S. Schwertfeger, and K. Pathak. "A networking framework for teleoperation in safety, security, and rescue robotics". In: *IEEE Wireless Communications* 16.1 (2009), pp. 6–13. DOI: 10.1109/MWC.2009.4804363.

[13] A. Tikanmäki, T. Bedrník, R. Raveendran, and J. Röning. "The remote operation and environment reconstruction of outdoor mobile robots using virtual reality". In: *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2017, pp. 1526–1531. DOI: 10.1109/ICMA.2017.8016043.

[14] B. Jincun, L. Qi, and L. Yanfei. "The design of the rescue robot long-distance control based on 3G and GPS". In: *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Vol. 2. 2009, pp. 170–172. DOI: 10.1109/IHMSC.2009.166.

[15] P.M. Duong, T.T. Hoang, N.T.T. Van, D.A. Viet, and T.Q. Vinh. "A novel platform for internet-based mobile robot systems". In: *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. 2012, pp. 1972–1977. DOI: 10.1109/ICIEA.2012.6361052.

[16] M.S. Uddin, M. Gianni, and A. Lab. "Long range robot teleoperation system based on Internet of Things". In: *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*. 2017, pp. 163–167. DOI: 10.1109/CCOMS.2017.8075288.

[17] X. Shen, Z.J. Chong, S. Pendleton, G.M.J. Fu, B. Qin, E. Frazzoli, and M.H. Ang. "Teleoperation of on-road vehicles via immersive telepresence using off-the-shelf components". In: *Intelligent Autonomous Systems 13*. Ed. by E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi. 2016, pp. 1419–1433. DOI: 10.1007/978-3-319-08338-4_102.

[18] P. Sossalla, J. Rischke, G.T. Nguyen, and F.H.P. Fitzek. "Offloading robot control with 5G". In: *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. 2022, pp. 461–464. DOI: 10.1109/CCNC49033.2022.9700709.

[19] L. Vignaroli, M. Gramaglia, M. Fuentes, A. Casella, R. Odarchenko, L. Natale, B. Altman, and F. D'Andria. "The touristic sector in the 5G technology era: the 5G-TOURS project approach". In: *2020 IEEE Globecom Workshops (GC Wkshps*. 2020, pp. 1–6. DOI: 10.1109/GCWkshps50303.2020.9367418.

[20] J. Marquez-Barja, S.A. Hadiwardoyo, V. Maglogiannis, et al. "Enabling cross-border tele-operated transport in the 5G era: the 5G Blueprint approach". In: *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. 2021, pp. 1–4. DOI: 10.1109/CCNC49032.2021.9369619.

[21] H. Kim. "5G vertical trials, use cases and scenarios". In: *2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*. 2022, pp. 81–86. DOI: 10.1109/IoTaIS56727.2022.9975928.

[22] C.C. Lessi, A. Gavrielides, V. Solina, R. Qiu, L. Nicoletti, and D. Li. "5G and beyond 5G technologies enabling industry 5.0: network applications for robotics". In: *Procedia Computer Science* 232 (2024), pp. 675–687. DOI: 10.1016/j.procs.2024.01.067.

[23] S. Profanter, A. Tekat, K. Dorofeev, M. Rickert, and A. Knoll. "OPC UA versus ROS, DDS, and MQTT: performance evaluation of Industry 4.0 protocols". In: *2019 IEEE International Conference on Industrial Technology (ICIT)*. 2019, pp. 955–962. DOI: 10.1109/ICIT.2019.8755050.

[24] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and M. Stiemerling. *Real-time streaming protocol version 2.0*. RFC 7826. 2016. DOI: 10.17487/RFC7826.

[25] B. Sredojev, D. Samardzija, and D. Posarac. "WebRTC technology overview and signaling solution design and implementation". In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015, pp. 1006–1009. DOI: 10.1109/MIPRO.2015.7160422.

[26] N. Slamnik-Kriještorac and J.M. Marquez-Barja. *5G-Blueprint whitepaper: next-gen teleoperation – 5G applications in modern ports*. Tech. rep. 5G-Blueprint Consortium, 2024. URL: https://www.5gblueprint.eu/wp-content/uploads/sites/62/2024/06/5G-Blueprint-White-Paper-Next-Gen-Teleoperation.pdf.